ABSTRACT

Title of Thesis: POST-FAILURE TRAJECTORY PLANNING FROM FEASIBLE TRIM STATE SEQUENCES

Matthew James Strube, Master of Science, 2005

Thesis directed by: Assistant Professor Ella M. Atkins
Department of Aerospace Engineering

Although today's aircraft provide a safe and reliable form of transportation, in this era of stringent safety requirements and increased hostile threat at home and abroad, accidents do occur. In the event of an emergency, rapid and precise action is required to avoid the loss of aircraft, crew, and any potential passengers. Some of the most difficult emergencies to manage are those that alter or reduce aircraft performance. When such failures occur, aircraft control can become more complex, requiring in some cases the pilot to re-learn how to fly. Moreover, once these new dynamics are learned, the pilot must effectively utilize them to ensure a safe landing. Providing this capability has been the goal of many researchers as they improve aircraft avionics and mechanical systems, although work done to develop emergency flight planners for reduced performance aircraft has been lacking.

This thesis presents a general method of autonomously generating emergency flight trajectories for post-failure aircraft connecting the aircraft with a desired

landing site. This emergency flight planner utilizes a simplified aircraft kinematic model allowing rapid computation of aircraft configuration changes from a sequence of trimmed, i.e., non-accelerating, flight conditions. The complete set of attainable trimmed flight conditions yields an accurate approximation of the post-failure flight envelope, guaranteeing the production of feasible flight plans. To facilitate accurate results, the feasibility and configuration impact of the dynamic transitions between these trim states must also be addressed. The flight planner uses a combination of discrete search and local continuous optimization techniques to piece together from compiled trim and transition databases, finding the necessary flight segment durations that produce the desired feasible flight trajectory to a known desired landing site. A case study focusing on lateral actuator (aileron and rudder) jams of an F-16 aircraft is used to demonstrate flight planner performance.

POST-FAILURE TRAJECTORY PLANNING FROM FEASIBLE TRIM STATE SEQUENCES

by

Matthew James Strube

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland at College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2005

Advisory Committee:

Assistant Professor Ella M. Atkins, Chairperson/Co-Advisor
Associate Professor Robert M. Sanner, Co-Advisor
Assistant Professor Benjamin Shapiro

DEDICATION

To Sarah and my parents.

ACKNOWLEDGEMENTS

Thanks, first, to Igor who did much of the leg work putting together the ideas that fueled this work and who acted the European spy by suppling "secret internal documents" from Airbus.

This work would not have been possible without the continued support of NASA Ames Research Center who provided the goals, and the money.

Thanks to the staff, faculty and students of the Space Systems Laboratory for the great learning and working environment. Special thanks go out to J. Bizzle and Meghan B. who were there when I needed them.

Thanks to my parents and close friends that told me that I could be anything I wanted to be. Whether its an engineer or not, the jury is still out. I'll always have my list...

Thanks to my advisers Ella and Rob, for their guidance and, more importantly, their patience.

Lastly, thanks to Sarita, who has always known me as a poor, stressed college student. Without your constant encouragement and support, Lord knows where I would be.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

On November 22, 2003, an Airbus A300 took off from Baghdad International Airport [1]. The aircraft, owned and operated by DHL International, was bound for Bahrain to deliver a load of mail from the Iraqi capital. Due to the operational hazards inherent to flying in and out of a war zone, DHL flight crews were instructed to take special operational precautions during take-off and landing; the new procedures were aimed at maximizing the take-off climb rate to limit the duration the aircraft would be vulnerable to potential ground threats. Despite their best efforts, the flight crew felt the force of an explosion as a rocket impacted the aircraft, removing a significant portion of the out board section of the left wing.

The result of the missile impact was the total loss of hydraulic power to all control surfaces: the ailerons, elevator, and rudder were all free floating, the spoilers were inoperative, and the flaps and slats were frozen. However, the two under-wing mounted engines were still running. Such a configuration allowed the aircraft to stay aloft long enough for the flight crew to learn the new dynamics of the aircraft resulting from the missile impact. After gaining some experience, the flight crew understood how they could alter the engine thrust to produce both

Figure 1.1: Damaged DHL Airbus A300

Source: Rosay [1]

pitch control, by altering the thrust symmetrically, and roll control, utilizing differential thrust. Using only the engines, the flight crew were able to successfully pilot the aircraft back to the airport and land the aircraft under controlled conditions.

1.1 Motivation

Although today's aircraft provide a safe and reliable form of transportation, in this era of stringent safety requirements and increased hostile threat at home and abroad, accidents do occur. In the event of an emergency, fast and precise action is required to avoid the loss of aircraft, crew, and any potential passengers. Providing this capability has been the goal of many researchers as they seek to improve aircraft avionic and mechanical systems to not only reduce the

probability of such incidents, but also provide pilots with the ability to manage emergencies when they arise.

Some of the most difficult emergencies to manage are those that result in altered or reduced aircraft performance. When such a failure occurs, the aircraft becomes more difficult to control, and in some cases, requires the pilot to re-learn how to fly. Moreover, once these new dynamics have been learned, it is critical that the pilot use them to make efficient and accurate decisions to ensure a safe landing. In A300 incident, the altered flight envelope—the feasible maneuver suite of the post-failure vehicle—was drastically different than the one the pilots were trained with. However, the new dynamics were such that the pilots could retain stable control of the aircraft as well as plan a feasible trajectory to the nearest runway. While in this case, the pilots were able to successfully land the aircraft, many emergency situations result in aircraft loss due to untrained pilots, or even more severe failures that the flight crew would be incapable of compensating for. As a result, the use of advanced avionics must be able to supply the pilot with the necessary additional information for landing the aircraft, or, in the most severe cases, automatically perform the required flight planning and execution activities.

The goal of this thesis is to present a general method of autonomously constructing emergency flight trajectories for reduced performance aircraft connecting the aircraft with a desired landing site. To validate the planning method, this thesis presents a case study showing implementation of the planner on an F-16 under varying degrees of lateral actuator—rudder and aileron—failure.

1.2 Approach

This thesis has two main thrusts. First, a simplified aircraft model is developed allowing rapid computation of aircraft configuration (position and orientation) changes in inertial space. This model builds sequences of trimmed, i.e., unaccelerated, segments and the appropriate dynamic transitions connecting them. The complete set of attainable trimmed flight conditions yields an accurate approximation of the post-failure flight envelope, guaranteeing the production of feasible flight plans for nominal or reduced performance conditions. The kinematic properties of these trimmed flight conditions allows the derivation of an analytical expression for the configuration change accumulated while maintaining the flight condition for a nonzero length of time.

This simplified aircraft model requires a detailed dynamic analysis to determine the feasibility and impact of transitioning between trimmed flight conditions. Indeed, it is shown that the natural dynamics of the aircraft prohibit the use of open-loop control techniques. As a result, a nonlinear controller, designed using techniques from linear systems theory, is presented, as well as a method of computing the flight path change incurred to build each transition.

Second, a flight planning algorithm is described to build sequences of these trimmed flight conditions into a valid trajectory connecting some initial aircraft location with a desired landing site. This flight planner uses a combination of discrete search and continuous local optimization techniques to piece together elements from trim and maneuver transition databases, finding the necessary durations to hold each trim segment to produce an acceptable flight plan.

Chapter 2 presents the rigid-body aircraft equations of motion as well as specific model information to simulate actuator failures of an F-16 aircraft. Chapter

3 rigorously defines trimmed flight and shows the creation of a trim database that approximates the continuous flight envelope. Chapter 3 also defines the simplified aircraft model by solving the aircraft kinematic equations of motion during a trimmed flight segment and shows how a flight plan of sequenced trimmed flight conditions can be modeled via simple matrix transformations. Chapter 4 examines the dynamic motion of the transitioning aircraft and presents a design of a nonlinear controller that allows accurate mapping of the overall flight path change during a transition. Chapter 5 defines the flight planning algorithm and the reduction of the initial trim database of Chapter 3 used to limit the total search space. Chapter 6 presents a case study examining flight planner efficiency and landing trajectory characteristics for a variety of rudder and aileron jam scenarios for an F-16 aircraft. This thesis concludes with a summary and directions for future work in Chapter 7.

1.3 Background

1.3.1 Emergency Flight Management Systems

Today's flight management systems (FMS) [2, 3] are capable of controlling the aircraft from take-off through landing so long as nominal flight conditions exists. However, in cases where a damaged aircraft responds differently than the nominal reference model, these systems are challenged to adapt appropriately. As a result, researchers have begun designing flight management architectures capable of effectively assisting the pilot during emergencies. Emergency flight planners (EFP) [4, 5] have shown the potential to reduce pilot errors provided accurate presentation of data or provide feasible post-failure flight plans for implementation by an autopilot. Similar architectures have also been developed for unmanned aerial

vehicle (UAV) applications [6] as well as emergency flight planning algorithms for specific failures, such as the engine out scenario [5, 7].

While development of EFP-type architectures is relatively new, previous research has focused on human factors aspects [4] of flight management systems, that is, analyzing the pilot's situational awareness and the impact various levels of automation can play in effective decision making. Research has also focused on the use of adaptive control techniques, coupled with advanced system identification modules [6], to compensate for changes in the nominal FMS performance model. While most researchers focus on one primary topic, EFP-type architectures typically include a representation of the following flight planning modules: an automatic plan generator, a trajectory predictor, autopilot, pilot interface, and system identification tools. The goal of the combined system is to produce a feasible waypoint sequence that the pilot or autopilot can navigate to a desired landing site.

A general EFP-type architecture is presented in Figure 1.2 [7, 8] and shows the interface of the research contributions of this thesis highlighted in yellow. At the highest level, EFPs normally contains a variable autonomy pilot interface and flight plan monitor that propagates the currently executing flight plan through the post-failure performance model. If the flight plan is deemed infeasible or unsafe, the pilot is notified and the adaptive flight planner (AFP) is activated. The AFP has two tasks: selecting a safe landing site, ideally an open runway, and planning a post-failure trajectory to that runway via the adaptive trajectory planner.

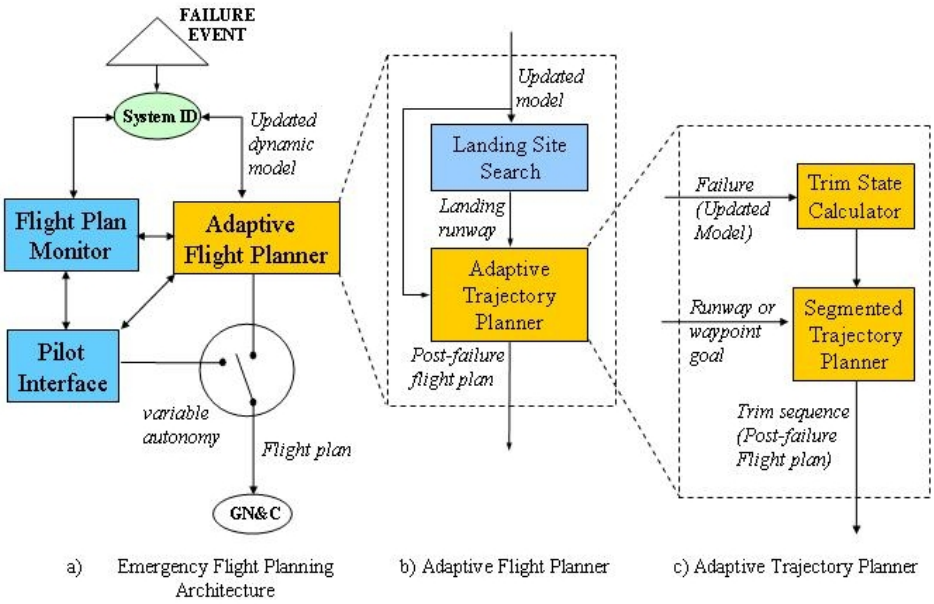


Figure 1.2: Post-Failure Trajectory Planning in Context of an Emergency Flight Planner

1.3.2 Adaptive Flight Control

An emergency flight planner is only one component of the overall management structure of an in-flight failure. It is essential that the controller, pilot or autopilot, maintain stable flight at all times, avoiding conditions outside the post-failure flight envelope. Much research has been conducted in maintaining aircraft stability for a variety of failure situations. In particular, adaptive controllers have been implemented to compensate for control surface failures [9–11], as well as airframe icing [12]. Adaptive critics [11, 13] have also been shown to improve piloting ability during an emergency by adjusting dynamic parameters inside the reference model so long as the pilot commanded trajectories are still within the flight envelope. Intelligent flight controllers [14, 15], augmented with a reference model, have enabled pilots to maintain control of a damaged aircraft following

extensive failures on a C-17 aircraft with redundant control surfaces. Such post-failure flight envelope protection is crucial for any autopilot or FMS to maintain stable flight.

1.3.3 Emergency Flight Planning

Producing feasible flight plans has been a topic of considerable research during the recent decade, but these efforts have typically considered nominal aircraft performance characteristics. UAV researchers have developed flight management tools more directed at fully-autonomous operation. In particular, Boskovic and Mehra [6, 16] define a layered control architecture consisting of modules for strategic decision making, tactical planning, and reconfigurable flight control analogous to a piloted flight management model combining the pilot, emergency flight planner, and flight controller. In their work, Boskovic and Mehra construct a set of alternate routes offline to respond to anomalous events, handling the set of most probable emergency situations that might otherwise require extensive deliberation to handle given a complex battle scenario. Similarly, Schouwenaars, Mettler, et al. [17–19] have applied dynamic programming to a minimal time-to-go cost function in order to dynamically define UAV flight plans from a database of trim conditions and maneuvers. Such techniques require a non-trivial planning cycle at each time step which necessitates the generation of rescue paths, such as a loiter or holding pattern, to buy planning time when unexpected events occur. In addition, Tomlin et al. [20] have devised a provably-correct real-time algorithm to guarantee two-aircraft collision avoidance and have evaluated it as a tool for the pilot and air traffic controllers.

1.3.4 Trajectory Generation

Much of the literature on automatic aircraft trajectory generation uses elements of optimal control theory to develop continuous state-space solutions that minimize fuel and time subject to airspace and air traffic constraints. Betts [21] presents a thorough review of two-point boundary value problems with direct and indirect solution techniques. Seywald et al. [22, 23] and Schultz [24] discuss trajectory optimization for aircraft flying in the longitudinal plane using a point mass performance model. Slattery and Zhao [25] synthesize trajectories for air traffic management to enable controllers to better guarantee safety and increase efficiency via minimal spacing. Wu and Guo [26] optimize trajectories based on total energy control over the climb, cruise, and descent phases of flight; the climb and descent profiles are produced by integrating the equations of motion with total energy as the independent variable.

Pilot-preferred commercial and general aviation (GA) flight plans are typically defined by a sequence of waypoints connected by constant-trim segments and transitions between these trim states. Such segmented routes [27] enable intuitive comprehension by pilots and ATC, facilitate communication of the trajectory, and can reduce computational complexity relative to numerical optimization processes. Frazzoli [28–30] used this concept to develop a hybrid automaton model of aircraft motion through a *quantization* of the system dynamics; Frazzoli restricts vehicle motion to trajectories of time-parameterized trimmed flight segments (what he calls *motion primitives*) connected by appropriately defined elements from a maneuver library. While he is mostly concerned with proving certain qualities such as well-posedness, consistency, and reachability for hybrid aircraft automaton, Frazzoli has also applied this method to highly maneuverable

helicopter motion planning.

This thesis describes an application of Frazzoli's hybrid framework on a less maneuverable platform, a conventionally controlled aircraft, as well as an extension of his motion primitives to a heterogenous atmosphere. Rather than prove hybrid automaton properties for fixed-wing aircraft, this work instead focuses on building trim sequences for the emergency flight planning problem.

Chapter 2

Aircraft Flight Model

This chapter presents the set of nonlinear differential equations that define the motion of a rigid aircraft operating in three-dimensional space. The union of these equations of motion with detailed aerodynamic and propulsion data for a specific vehicle fully characterize that vehicle's post-failure performance. Section 2.1 presents the rigid-body vector equations of motion for a six-degree-of-freedom vehicle as well as a description of the variables used, coordinate reference frames, and the underlying assumptions. Section 2.2 describes an alternate velocity reference frame commonly used by the aerospace industry. Section 2.3 discusses connecting the general equations of motion to specific vehicles through the use of aerodynamic and propulsion databases containing the necessary force and torque information, as well as the specific F-16 aircraft model used for this work. Finally, Section 2.4 presents a compact representation for the equations of motion.

2.1 Nonlinear Equations of Motion

2.1.1 Aircraft Dynamics

The dynamics of a rigid aircraft [31] can be expressed in vector form as

$$ {}^{B}\dot{\mathbf{v}} = -\boldsymbol{S}({}^{B}\boldsymbol{\omega}){}^{B}\mathbf{v} + \boldsymbol{R}_{BI}{}^{I}\mathbf{g} + \frac{{}^{B}\boldsymbol{F}}{m} \tag{2.1} $$

$$ {}^{B}\dot{\boldsymbol{\omega}} = -\boldsymbol{H}^{-1}\boldsymbol{S}({}^{B}\boldsymbol{\omega})\boldsymbol{H}{}^{B}\boldsymbol{\omega} + \boldsymbol{H}^{-1}{}^{B}\boldsymbol{T} \tag{2.2} $$

where ${}^{B}\mathbf{v}$ and ${}^{B}\boldsymbol{\omega}$ are three dimensional vectors defining respectively the linear and angular velocity of the system. The preceding superscript denotes that the velocities are measured in the aircraft body coordinate B frame which is rigidly embedded at the center of gravity of the vehicle and aligned so that the x-axis is pointed out the vehicle's nose, the y-axis is pointed down the right wing, and the z-axis projects out the underside of the aircraft. ${}^{B}\boldsymbol{F}$ and ${}^{B}\boldsymbol{T}$ are also measured in the aircraft body frame and respectively define the vector sum of the three dimensional aerodynamic and propulsion forces and torques acting on the aircraft. $\boldsymbol{H}$ is a matrix representing the inertia properties of the system, while m is the mass of the vehicle. $\boldsymbol{R}_{BI}$ is the rotation matrix that converts vectors expressed in the inertial frame to components in the aircraft body frame and ${}^{I}\mathbf{g}$ is the influence of gravity on the system measured in inertial (I) coordinates. Finally, the matrix operator $\boldsymbol{S}$ is a matrix representation of the cross product. For example, the cross product $\mathbf{x} \times \mathbf{y}$ can be represented as

$$ \mathbf{x} \times \mathbf{y} = \boldsymbol{S}(\mathbf{x})\mathbf{y} $$

where $\boldsymbol{S}$ is the skew-symmetric matrix

$$\boldsymbol{S}(\mathbf{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

Sometimes, it will be desirable to address the individual components of the vectors ${}^{B}\mathbf{v}$ and ${}^{B}\boldsymbol{\omega}$ which are conventionally expressed as

$${}^{B}\mathbf{v}^{T} = [U,\ V,\ W]$$

$${}^{B}\boldsymbol{\omega}^{T} = [p,\ q,\ r].$$

Because B is aligned along the aircraft's lines of symmetry, p, q, and r are often called the roll, pitch, and yaw rates, respectively.

2.1.2 Aircraft Kinematics

Aircraft position is expressed by the three dimensional vector $\mathbf{p}$ which locates the center of the aircraft body frame with respect to an inertial reference frame. The origin of the inertial coordinate frame can be placed arbitrarily on the surface of the Earth, as long as it is fixed, and is traditionally oriented such that the x and y axes are aligned respectively with the North and East cardinal directions, and the z-axis is directed downward. With this definition of the inertial coordinate frame, the aircraft position vector obeys the kinematic relation

$$\dot{\mathbf{p}} = \boldsymbol{R}_{BI}^{T}\,{}^{B}\mathbf{v} \tag{2.3}$$

which shows how the velocity, measured in body frame B, results in a change in position, measured in inertial frame I.

$\boldsymbol{R}_{BI}$ is the rotation matrix that converts vectors measured in the inertial frame to aircraft body frame components. Representing a rotation in three dimensional

space, it can be expressed more compactly as a sequence three planar rotations about the aircraft body axes. Therefore, for any vector ${}^{I}\mathbf{v}$, the coordinate transformation rotating vectors between the inertial to the aircraft body frame can be expressed as

$$ {}^{B}\mathbf{v} = \boldsymbol{R}_{\phi}\, \boldsymbol{R}_{\theta}\, \boldsymbol{R}_{\psi}\, {}^{I}\mathbf{v} \tag{2.4} $$

where

$$ \begin{gathered} \boldsymbol{R}_{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \qquad \boldsymbol{R}_{\theta} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, \\ \boldsymbol{R}_{\psi} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{gathered} \tag{2.5} $$

The angles ϕ, θ, ψ are referred to as, respectively, the roll (or bank), pitch, and yaw angles and correspond to the body-axis angular rates p, q, r. Roll represents wing tilt, while pitch represents nose angle with respect to the horizon—the inertial x-y plane. These angles have values in the interval $[-\pi, \pi]$, where positive angles match right-handed rotations about the aircraft x and y body axes. The yaw angle is the angle the nose makes with the inertial x direction, measured counter-clockwise when viewed along the negative inertial z-axis, and takes on values in the interval $[0, 2\pi]$. This three-variable attitude description is known as the Euler representation where ϕ, θ, ψ are the Euler angles. Using the above representation for the aircraft orientation yields the following kinematic relationship

between the Euler angles, their rates, and the body-axes angular rates:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{2.6}$$

This relationship can also be expressed in vector form as

$$\dot{\mathbf{\Phi}} = \boldsymbol{E}(\mathbf{\Phi})^B\boldsymbol{\omega} \tag{2.7}$$

where $\mathbf{\Phi}^T = [\phi,\ \theta,\ \psi]$.

2.1.3 Modeling Assumptions

Together, (2.1), (2.2), (2.3), and (2.7) are known as the rigid, flat Earth, body-axes equations of motion for a six-degree of freedom system. The most important assumption used in deriving the above equations is that the aircraft is a rigid body. An aircraft is anything but rigid, which is not a surprise to any air traveler who has observed the wings of their aircraft flex and bend throughout a flight. Adding the appropriate equations accurately modeling this flexure would severely complicate the equations of motion, but, in actuality, only minimally affects the position and orientation states. The influence of these flexible dynamics are neglected in the above equations.

The other main assumption is that the Earth is flat. The Earth is actually an ellipsoid and for accurate around-the-Earth navigation, this curvature must be accounted for as well as the slow rotation of the Earth about its axis. In this work presented here, it is assumed that the lengths of the paths planned are negligible compared to the radius of the Earth and that compensating for such errors would not significantly improve accuracy. It should also be noted that

the flight planning methodology presented here takes no major liberties with this assumption and can be appropriately modified to use the more traditional latitude and longitude parameterization if desired.

Another consideration is the use of Euler angles to define aircraft orientation. One side-effect of using a three-variable attitude representation is the existence of a singularity in the orientation kinematics. Specifically, a singularity exists when the pitch angle reaches $\pi/2$; this results in an area around the singularity where a simulation will experience numerical instability. In this work, practical limits on aircraft performance prevent the singularity from being reached in nominal and post-failure scenarios.

2.2 Wind Axes Coordinates

The equations of motion provided above make use of an aircraft body reference frame to define aircraft linear velocity. Aerodynamicists, however, have traditionally used an alternative representation parameterizing linear velocity in terms of vehicle orientation with respect to the relative wind vector. When an aircraft moves through the air, the airflow over the body is uniform and, as such, can be defined by the free-stream velocity v, also known as the airspeed, and two angles describing the direction of the oncoming wind, the angle-of-attack α and side slip angle β.

Figure 2.2 shows α and β in terms of the wind vector and B frame. The angle of attack describes how much "nose-up" orientation the aircraft with respect to the wind and the side slip angle describes the angle the wind makes with the center line of the aircraft. The angle of attack and side slip are the angles by which, respectively, the aircraft B frame must be rotated to align the wind vector

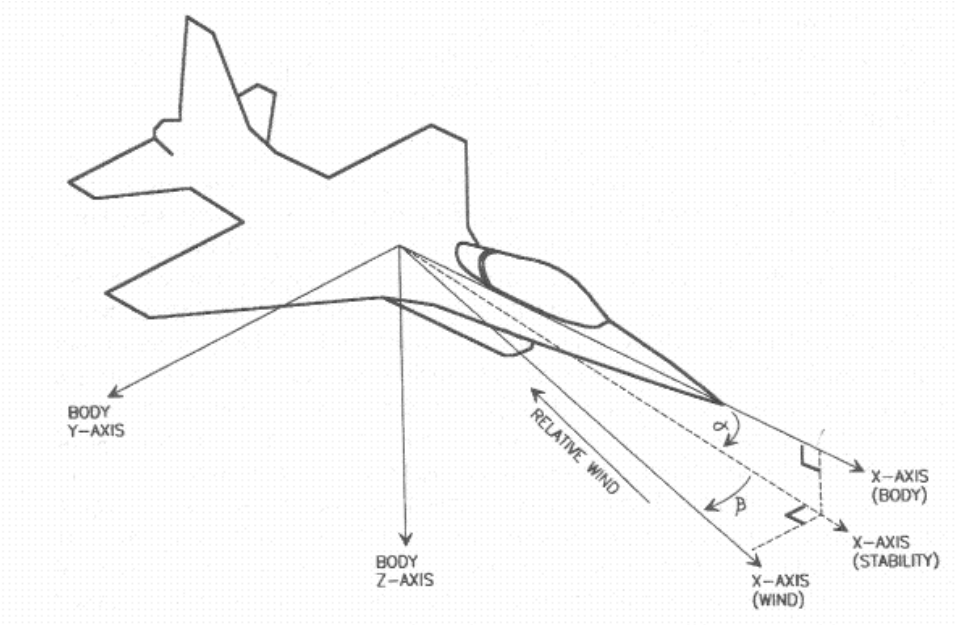


Figure 2.1: Definition of Wind Axes and Angles

Source: Stevens & Lewis [31]

with the negative x body axis, defining the new W (for "wind") reference frame. As a result, for a given ${}^{B}\mathbf{v}$,

$$\begin{aligned} {}^{W}\mathbf{v} &= \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} {}^{B}\mathbf{v} \\ &= \boldsymbol{R}_{\beta}\, \boldsymbol{R}_{\alpha}\, {}^{B}\mathbf{v} \\ &= \boldsymbol{R}_{WB}\, {}^{B}\mathbf{v} \end{aligned} \tag{2.8}$$

where ${}^{W}\mathbf{v}^{T} = [v,\ 0,\ 0]$. Expanding (2.8) yields the explicit definitions of the

aerodynamic variables in terms of the linear velocity measured in the B frame.

$$\begin{aligned} v &= (U^2 + V^2 + W^2)^{1/2} \\ \tan\alpha &= \frac{W}{U} \\ \sin\beta &= \frac{V}{(U^2 + V^2 + W^2)^{1/2}}. \end{aligned} \tag{2.9}$$

Differentiating (2.9) leads to the following differential equations for v, α, β:

$$\begin{aligned} \dot{v} &= \frac{U\dot{U} + V\dot{V} + W\dot{W}}{v} \\ \dot{\alpha} &= \frac{U\dot{W} - W\dot{U}}{U^2 + W^2} \\ \dot{\beta} &= \frac{\dot{V}v - V\dot{v}}{v^2\cos\beta} \end{aligned} \tag{2.10}$$

which can be used in conjunction with (2.1) to completely define aircraft dynamics.

2.3 Aerodynamic Forces and Moments

While the governing equations of motion for a rigid vehicle in three dimensional space are the same for all aircraft, specific vehicles differ in the aerodynamic forces ${}^{B}\boldsymbol{F}$ and torques ${}^{B}\boldsymbol{T}$ acting on them. In order to build a precise aircraft performance model, a detailed aerodynamic study of each aircraft type must be formed. This mathematical model can be used for any nominal and emergency path planning.

2.3.1 Aerodynamic and Propulsion Effects

Aircraft forces and torques are primarily governed by aerodynamic effects and engine thrust and can be expressed as

$$\begin{aligned} {}^{B}\boldsymbol{F} &= {}^{B}\boldsymbol{F}_{A} + {}^{B}\boldsymbol{F}_{T} \\ {}^{B}\boldsymbol{T} &= {}^{B}\boldsymbol{T}_{A} + {}^{B}\boldsymbol{T}_{T} \end{aligned} \tag{2.11}$$

where subscripts A and T denote the influence of aerodynamic effects and engine thrust respectively. ${}^{B}\boldsymbol{F}_{A}$ and ${}^{B}\boldsymbol{T}_{A}$ are typically determined through wind tunnel tests on scale models of the aircraft, and in many cases are supplemented by actual flight test data. These aerodynamic effects are highly dependent on aircraft velocity, both linear and angular, as well as atmospheric characteristics given by the altitude at flight altitude. ${}^{B}\boldsymbol{F}_{T}$ and ${}^{B}\boldsymbol{T}_{T}$ are determined by the engine manufacturer and also depend on both aircraft speed and altitude.

Aircraft control is provided through actuators on aerodynamic surfaces, altering the airflow over individual lifting surfaces on the vehicle, thereby effecting the magnitude and direction of ${}^{B}\boldsymbol{F}_{A}$ and ${}^{B}\boldsymbol{T}_{A}$. Similarly, the pilot has complete authority over ${}^{B}\boldsymbol{F}_{T}$. These actuator and engine settings are combined to form $\boldsymbol{\mu}$ an m dimensional vector of control inputs to the system.

2.3.2 F-16 Aircraft Force & Torque Data

For this work, the mathematical model for ${}^{B}\boldsymbol{F}$ and ${}^{B}\boldsymbol{T}$ are based on aerodynamic data from NASA-Langley wind tunnel tests on a scale model F-16 aircraft [31] designed to explore stall and post-stall regions of a relaxed stability aircraft. As a result, the aerodynamic data is valid for speed ranges up to Mach 0.6, angle-of-attacks ranging from -10° to 45°, and side slip angles in the range of $\pm 30^{\circ}$. The

thrust model is based on the F-16 after-burning engine and neglects the spool-up time of the engine rotors.

The control input $\boldsymbol{\mu}$ for the F-16 consists of elevator, aileron, and rudder deflections, together with the thrust or throttle setting:

$$\boldsymbol{\mu} = [\mu_t,\ \mu_e,\ \mu_a,\ \mu_r]^T$$

where the subscript on each scalar μ denotes the particular control setting. The range of possible inputs $\boldsymbol{\mu}$ are limited by the physical properties of the system. For the F-16, the elevators μ_e are limited to deflections of $\pm 25°$, the ailerons μ_a to $\pm 21.5°$, and the rudder μ_r to $\pm 30°$. However, the control models neglect the lag between command and actual deflections. The throttle input μ_t is often normalized with respect to the maximum setting so that the corresponding feasible values it can take on are constrained to lie between zero, no thrust, and 1, maximum thrust.

To simulate failures using the available data, the only failure modes considered in this work are "jammed" actuators where one of the three aerodynamic actuators will be stuck at some constant deflection. Such a failure will influence ${}^B\boldsymbol{F}$ and ${}^B\boldsymbol{T}$ while preventing the pilot from directly controlling the failed actuator. In particular, the results presented in this work will focus on rudder and aileron jam failure scenarios. Thus, the control vector $\boldsymbol{\mu}$ used is three dimensional and can be expressed as

$$\boldsymbol{\mu}^T = [\mu_t,\ \mu_e,\ \mu_a \text{ or } \mu_r]$$

where the use of either μ_a or μ_r will be specified explicitly based on the failure case. With this representation, ${}^B\boldsymbol{F}$ and ${}^B\boldsymbol{T}$ must contain the aerodynamic contribution of the failed rudder or aileron at its jammed setting.

While the method above will be used to develop specific rudder and aileron failure models emergency flight planning discussed in this work, the fundamental method places no restrictions on the type of failure. However, the specific method described in this work requires at least three post-failure controllable actuators. Aerodynamic data modeling failed actuators, reduced authority actuators, missing wings and other structural damage, etc. can also be used in conjunction with the flight planner and controller combination developed in this thesis, provided this minimum controllable actuator criterion is met. Removing this restriction would require the development of new feedback control law that would work given a more limited set of remaining actuators.

2.4 State Representation

Note that (2.1), (2.2), (2.3), and (2.7) can be more compactly represented as

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \boldsymbol{\mu}) \tag{2.12}$$

The 12-dimensional state vector $\mathbf{z}$ can be expressed in a partitioned form as

$$\mathbf{z}^T = [\, \boldsymbol{\eta}^T \; \boldsymbol{\nu}^T \,]$$

where

$$\boldsymbol{\eta}^T = [\, x,\; y,\; h,\; \phi,\; \theta,\; \psi \,]$$

is the 6-dimensional *configuration* of the vehicle—its position and attitude—while

$$\boldsymbol{\nu}^T = [\, v,\; \alpha,\; \beta,\; p,\; q,\; r \,]$$

fully defines the linear and angular *velocities* of the aircraft. Here, altitude h replaces inertial z to conform with traditional aerospace notation, and can be

used interchangeably through the relation

$$h = -z.$$

An alternate, and equivalent, view of the state vector is that it completely defines the stored energy of the system. The configuration $\boldsymbol{\eta}$ specifies the stored potential energy, while the velocities $\boldsymbol{\nu}$ naturally describe the system's kinetic energy. As the aircraft moves through its environment, the equations of motion describe the flow of energy through the system, requiring that the state vector update to reflect changes. Knowledge of the state vector at any initial time and the applied control inputs from that time forward, completely defines aircraft motion, or state trajectory, which will expressed as the time varying state vector $\mathbf{z}(t)$.

Chapter 3

Trim Analysis

This chapter describes a discrete representation of the flight envelope that defines nominal and post-failure flight conditions to be assembled as landing trajectories. Computing this flight envelope requires the definition of a trim state—a non-accelerating flight condition—as well as a systematic procedure for trim state computation for any desired flight condition. Next, a simplified aircraft kinematic model is presented that determines the change in flight path (position and heading) due to a sequence of trimmed flight conditions. This model requires accurate characterization of the flight path change during a single trimmed flight segment as well as the compounded changes over a general sequence of feasible trim states.

Section 3.1 defines trimmed flight conditions, trim state families, and constant trimmed control settings computed with a constrained nonlinear optimization technique. In Section 3.2, aircraft stability and controllability are defined in small neighborhoods surrounding a trim state using results from linear systems theory. Test are presented that grade each trim state with respect to each property. This analysis leads into the definition of a trim database in Section 3.3 which is a discrete representation of the post-failure flight envelope. In Section

3.4, a set of equations are derived to express the evolution of the flight path parameters as a function of time, yielding the simplified aircraft kinematic model defined in Section 3.5. Finally, Section 3.6 presents a simple altitude-feedback trim controller that enables tracking of generic trimmed flight conditions.

3.1 Trimmed Flight Conditions

A trimmed flight condition occurs when an aircraft experiences an equilibrium stable non-accelerating flight. In a trim state, aircraft linear and angular velocities are constant and can be explicitly stated as a condition for which

$$\dot{\boldsymbol{\nu}}^* = 0 \tag{3.1}$$

or, equivalently, a flight condition for which $\boldsymbol{\nu}^* =$ constant, where an asterix will be used to denote an equilibrium quantity. Since the equations of motion are a function of not only the state of the system, but also the control input, as shown (2.12), trimmed flight conditions are also control dependent. In fact, different constant control settings can produce different trim states, so that more generally, $\boldsymbol{\nu}_k^*(\boldsymbol{\mu}_k^*)$ where $\boldsymbol{\nu}_k^*$ is the specific steady-state flight condition resulting from holding the constant control input $\boldsymbol{\mu}_k^*$.

3.1.1 Trimmed Climbing-Turning Flight

While the above condition (3.1) is necessary for all trimmed flight conditions, this work will requires steady climbing/descending turns as well as straight and level flight. To maintain these flight conditions, additional conditions must be placed on the required trim configuration rates $\dot{\boldsymbol{\eta}}$. For an aircraft to perform a

steady climbing/descending turn,

$$\begin{aligned} &\dot{v},\ \dot{\alpha},\ \dot{\beta},\ \dot{p},\ \dot{q},\ \dot{r} \equiv 0 \\ &\dot{\phi},\ \dot{\theta} \equiv 0 \\ &\dot{\psi} \equiv \dot{\psi}^* \\ &\dot{h} \equiv \dot{h}^*, \end{aligned} \tag{3.2}$$

where $\dot{h}^*$ and $\dot{\psi}^*$ are respectively the desired climb rate and turn rate for the trim condition. The climbing/descending turn is a more general flight condition for which level turning flight ($\dot{h} \equiv 0$), straight climbing/descending flight ($\dot{\psi} \equiv 0$), and straight and level flight ($\dot{h} \equiv 0$, $\dot{\psi} \equiv 0$) are all special instances. The suite of basic flight maneuvers performed by commercial and general aviation pilots can be represented by (3.2). Since trimmed flight requires aircraft pitch and roll to remain constant, a trim state can be fully defined by via the reduced state vector

$$\bar{\mathbf{z}}^T = [\phi,\ \theta,\ v,\ \alpha,\ \beta,\ p,\ q,\ r] \tag{3.3}$$

as $\bar{\mathbf{z}}_k^*(\boldsymbol{\mu}_k^*)$, where $\bar{\mathbf{z}}_k^*$ is the trim state arising from the constant control setting $\boldsymbol{\mu}_k^*$.

The trim values $\boldsymbol{\mu}_k^*$ and $\mathbf{z}_k^*$ can be found be solving $\dot{\bar{\mathbf{z}}}_k^* = 0$ for a specified $\dot{h}^*$ and $\dot{\psi}^*$. However, atmospheric density variation as a function of altitude means requires gradual control setting changes to maintain a trimmed flight condition. Each trim maneuver can also be executed over a range of airspeeds. Thus, there actually exists a family of trim state solutions given by

$$\begin{aligned} &\bar{\mathbf{z}}_k^*(h^*,\ v^*,\ \dot{h}^*,\ \dot{\psi}^*) \\ &\boldsymbol{\mu}_k^*(h^*,\ v^*,\ \dot{h}^*,\ \dot{\psi}^*) \end{aligned} \tag{3.4}$$

where h^* and v^* are respectively the altitude and forward airspeed of the desired trim state. Note that for climbing or descending flight, h^* is defined as the

initial altitude. The relatively slow variation of density allows the resulting constant trim computation to hold for a short amount of time after the start of the maneuver and subsequently be maintained by altering control surface positions.

3.1.2 Nonlinear Constrained Optimization

One method of computing the above family of trim solutions is to solve the related constrained nonlinear optimization problem that minimizes

$$J_{trim}(\mathbf{z},\ \boldsymbol{\mu}) = \frac{1}{2}\,\dot{\mathbf{z}}^T\,\mathbf{Q}\,\dot{\mathbf{z}} \tag{3.5}$$

subject to the equality constraint $\mathbf{g}(\mathbf{z},\ h^*,\ v^*,\ \dot{h}^*,\ \dot{\psi}^*) = \mathbf{0}$ and the inequality constraint $\mathbf{s}(\boldsymbol{\mu}) \leq \mathbf{0}$. $\mathbf{g}$ is a nonlinear function that enforces the commanded trimmed flight condition [31] and can be expressed by

$$\mathbf{g}(\mathbf{z},\ h^*,\ v^*,\ \dot{h}^*,\ \dot{\psi}^*) = \begin{bmatrix} h - h^* \\ v - v^* \\ \tan\theta - \dfrac{ab + \sin\gamma^*\sqrt{a^2 - \sin\gamma^{*2} + b^2}}{a^2 - \sin\gamma^{*2}} \\ p + \sin\theta\,\dot{\psi}^* \\ q - \cos\theta\sin\phi\,\dot{\psi}^* \\ r - \cos\theta\cos\phi\,\dot{\psi}^* \end{bmatrix} \tag{3.6}$$

where γ^* is the trimmed flight path angle defined as $v^*\sin\gamma^* = \dot{h}^*$, and

$$a = \cos\alpha\cos\beta$$

$$b = \sin\phi\sin\beta + \cos\phi\sin\alpha\cos\beta.$$

The first two constraints in (3.6) directly constrain the altitude and airspeed, whereas the specified climb rate is indirectly specified by the third constraint on the required flight path angle. Similarly, the turn rate is indirectly specified

through the last three constraints which are derived from the rotational kinematics and constrain the roll and pitch rates to be zero. The nonlinear function $\mathbf{s}$ describes the physical deflection limits on the actuators and for the F-16 can be expressed as

$$\mathbf{s}(\boldsymbol{\mu}) = \begin{bmatrix} \mu_t \\ 1 - \mu_t \\ 25 - |\mu_e| \\ 21.5 - |\mu_a| \text{ or } 30 - |\mu_r| \end{bmatrix} \tag{3.7}$$

The matrix $\boldsymbol{Q}$ can be any positive definite matrix and describes the relative weighting between the trim state derivatives so that each derivative contributes equally to (3.5).

When (3.5) is minimized with respect to the trimmed flight condition $(h^*, v^*, \dot{h}^*, \dot{\psi}^*)$, J^*_{trim} can be expressed as

$$J^*_{trim} = J_{trim}(\mathbf{z}^*,\ \boldsymbol{\mu}^*) = \min_{\mathbf{z},\mu}\{J_{trim}(\mathbf{z},\ \boldsymbol{\mu})\} \tag{3.8}$$

where $\mathbf{z}^*$ and $\boldsymbol{\mu}^*$ are the solution to the minimization. A trim state is considered feasible if

$$J^*_{trim} = 0. \tag{3.9}$$

The additional—non-trim—values in $\mathbf{z}^*$, namely x^*, y^*, and ψ^*, can be set to zero and ignored.

In practice, however, (3.5) can not be minimized analytically and a multi-dimensional numerical optimization algorithm must be used to compute an approximate solution. This algorithm will iteratively adjust a set of independent values until some internal, algorithm-specific solution criterion signals that J has reached a minimum. Variables the algorithm is able to independently adjust can

be found by examining (3.6) and are listed in Table 3.1, together with the remaining dependent variables and those that can be neglected in computing trim solutions.

Table 3.1: Numerical Trim Routine Variables

Independent	Dependent	No-Effect
α	h	x
β	v	y
ϕ	θ	ψ
μ_t	p	
μ_e	q	
μ_a or μ_r	r	

The feasibility condition (3.9) must be altered to accommodate numerical optimization. Specifically, the numerical trim algorithm cannot be expected to compute true zero-cost solutions due to numerical round-off error. As a result, a small positive scalar ϵ_{trim} can be defined such that if

$$J^*_{trim} < \epsilon_{trim} \ll 1 \tag{3.10}$$

then the flight condition can be considered feasible. Thus, ϵ_{trim} defines the largest trim cost a feasible flight condition is allowed to have and represents a numerical zero.

3.1.3 Feasible Flight Envelope

In solving (3.5), it becomes apparent that not every turn and climb combination is feasible at every altitude and airspeed due to the limited range of allowable

$\boldsymbol{\mu}$. Indeed, there will be maneuvers which cannot be performed regardless of the airspeed or altitude. These limitations define the *flight envelope* of the aircraft. More explicitly, the flight envelope is defined as the complete set of flight conditions (h^*, v^*, $\dot{h}^*$, $\dot{\psi}^*$) which satisfy (3.9) and therefore can be represented by a four-dimensional volume in the flight condition space. Thus, the flight envelope defines the complete feasible set of trimmed flight conditions available to the post-failure aircraft from which the flight planner will define feasible trajectories to a desired landing site.

3.2 Linear Trim Analysis

Besides feasibility, there are additional desired properties that trim states should possess. One such property is stability. A nonlinear system is considered stable, with respect to a trim state, if the system will naturally converge to the trim state if the state of the system is near the trim state [32]. When applying this definition to aircraft, the additional provision that the control input be fixed to the associated trim setting corresponding to the particular trim state leads to the more specific *stick-fixed* stability. Such a property is ideal when considering an aircraft is constantly influenced by wind gusts causing perturbations away from the trim linear and angular velocities. However, if these trim states are stable, these perturbations can be guaranteed to decay over time.

A more important and necessary property of trim states is that the system is controllable in the region surrounding a trim state. More specifically, a system is controllable if there exists an input that can transfer the aircraft between two distinct states in finite time [33]. For the purposes of this thesis, a more practical definition is that controllability implies that the actuators can be varied

in proportion to deviations in the aircraft state from trim in a manner which can stabilize the aircraft, even if the natural "stick-fixed" motion is unstable.

While a full analysis of nonlinear aircraft dynamics is daunting given the complex coupling of the state variables, it is impossible when the aerodynamic and propulsion data is supplied in tabular form, as with the F-16 model used in this work. Fortunately, a full nonlinear analysis is not needed to determine stability and controllability in regions around a trim state. In fact, an important result from nonlinear systems theory shows that, at least in a small neighborhood surrounding a trim state, the nonlinear system can be approximated by a linearization of its dynamics about that trim state [32]. The remainder of this section will develop this concept and use it to determine stability and controllability properties of aircraft trim states.

3.2.1 Linear Perturbation Models

Aircraft motion about each feasible trim point $\bar{\mathbf{z}}_k^*$ computed above maybe described by a linear differential equation:

$$\dot{\mathbf{x}}_k = \boldsymbol{A}_k \mathbf{x}_k + \boldsymbol{B}_k \mathbf{u}_k \tag{3.11}$$

where $\mathbf{x}_k = \bar{\mathbf{z}} - \bar{\mathbf{z}}_k^*$, $\mathbf{u}_k = \boldsymbol{\mu} - \boldsymbol{\mu}_k^*$, and $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are the constant Jacobian matrices

$$\boldsymbol{A}_k = \left.\frac{\partial \bar{\mathbf{f}}}{\partial \bar{\mathbf{z}}}\right|_{\bar{\mathbf{z}}=\bar{\mathbf{z}}_k^*,\, \boldsymbol{\mu}=\boldsymbol{\mu}_k^*} \tag{3.12}$$

$$\boldsymbol{B}_k = \left.\frac{\partial \bar{\mathbf{f}}}{\partial \boldsymbol{\mu}}\right|_{\bar{\mathbf{z}}=\bar{\mathbf{z}}_k^*,\, \boldsymbol{\mu}=\boldsymbol{\mu}_k^*} \tag{3.13}$$

where $\bar{\mathbf{f}}$ is the system of nonlinear equations defining the dynamics of $\bar{\mathbf{z}}$ at an altitude of h_k^*. The use of tabular aerodynamic and propulsion data complicates

an analytic derivation. However, in practice, $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ can be reasonably well approximated by first-order difference equations:

$$\begin{aligned} \boldsymbol{A}_{k,i} &\approx \frac{\bar{\mathbf{f}}(\bar{\mathbf{z}}_k^* + \epsilon_i \mathbf{e}_i, \boldsymbol{\mu}_k^*) - \bar{\mathbf{f}}(\bar{\mathbf{z}}_k^*, \boldsymbol{\mu}_k^*)}{\epsilon_i} \\ \boldsymbol{B}_{k,i} &\approx \frac{\bar{\mathbf{f}}(\bar{\mathbf{z}}_k^*, \boldsymbol{\mu}_k^* + \epsilon_i \mathbf{e}_i) - \bar{\mathbf{f}}(\bar{\mathbf{z}}_k^*, \boldsymbol{\mu}_k^*)}{\epsilon_i} \end{aligned} \tag{3.14}$$

where $\boldsymbol{A}_{k,i}$ is the i^{th} column of $\boldsymbol{A}$ and $\boldsymbol{B}_{k,i}$ is the i^{th}. ϵ_i is a sufficiently small, positive number and $\mathbf{e}_i$ is the i^{th} column of an n-dimensional identity matrix where n is the size of the corresponding $\mathbf{z}_k^*$ or $\boldsymbol{\mu}_k^*$.

3.2.2 System Stability

As a direct consequence of the equivalence of a nonlinear system and (3.11) near a trim state, the aircraft is asymptotically stable with respect to $\bar{\mathbf{z}}_k^*$ if

$$\Re\{\lambda_i(\boldsymbol{A}_k)\} < 0 \quad \forall i = 1, \ldots, 8 \tag{3.15}$$

where $\lambda_i(\boldsymbol{A})$ represents the i^{th} eigenvalue of $\boldsymbol{A}$ and small perturbations away from the k^{th} trimmed flight condition will decay asymptotically to zero [32].

As mentioned above, stable trim states are desirable since aircraft motion is inherently robust to small perturbations in the state variables. An unstable trim state, in which any one of the eigenvalues of $\boldsymbol{A}_k$ is positive, should be avoided since the aircraft will tend to diverge rapidly away from the trim state after experiencing any perturbation. Unstable trim states, however, may be *stabilizable*, i.e., they can be made stable, by appropriately coupling compensating actuator deflections to the difference between the actual state of the system and the desired trim state. Therefore, if the system is stabilizable, then by varying the actuator settings according to the control law $\mathbf{u}_k = -\boldsymbol{K}_k \mathbf{x}_k$, the closed-loop dynamics of

(3.11) can be shown to be

$$\dot{\mathbf{x}}_k = (\boldsymbol{A}_k - \boldsymbol{B}_k\boldsymbol{K}_k)\mathbf{x}_k. \tag{3.16}$$

which can then be made stable through an appropriate choice of $\boldsymbol{K}_k$.

3.2.3 System Controllability

Similarly, an aircraft is controllable in a neighborhood of $\bar{\mathbf{z}}_k^*$ if the matrix pair $[\boldsymbol{A}_k,\ \boldsymbol{B}_k]$ is controllable, which is equivalent to a controllability matrix

$$\boldsymbol{U}_C = \begin{bmatrix} \boldsymbol{B}_k & \boldsymbol{A}_k\boldsymbol{B}_k & \boldsymbol{A}_k^2\boldsymbol{B}_k & \dots & \boldsymbol{A}_k^{(n-1)}\boldsymbol{B}_k \end{bmatrix} \tag{3.17}$$

with full row rank [33]. More explicitly, the linear system in controllable if

$$\rho(\boldsymbol{U}_C) = n \tag{3.18}$$

where $\rho(\boldsymbol{A})$ represents the rank of the matrix $\boldsymbol{A}$ and n is the leading dimension—the number of rows—of both $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$. For the linear systems examined in this work, $n = 8$.

While stabilizability allows the closed-loop system matrix to be made Hurwitz, if the linear system is controllable, then the feedback control law $\mathbf{u}_k = -\boldsymbol{K}_k\mathbf{x}_k$ arbitrarily place the closed-loop eigenvalues. More specifically, if the matrix pair $[\boldsymbol{A}_k,\ \boldsymbol{B}_k]$ is controllable, then there exists a matrix $\boldsymbol{K}_k$ such that the eigenvalues of the closed-loop system (3.16) have the property

$$\lambda_i(\boldsymbol{A}_k - \boldsymbol{B}_k\boldsymbol{K}_k) = \lambda_{des,i} \quad \forall i = 1, \dots, 8 \tag{3.19}$$

where $\lambda_{des,i}$ is the i^{th} component of $\boldsymbol{\lambda}_{des}$, a vector containing the desired eigenvalues. Given a specific choice for $\boldsymbol{\lambda}_{des}$, a variety of algorithms exist to determine what $\boldsymbol{K}_k$ satisfies (3.19) [34]. The ability to completely control the aircraft in

a neighborhood surrounding the trim state $\bar{\mathbf{z}}_k^*$ is a much stronger property than stability or stabilizability. In future chapters, it will be shown that while stability can guarantee the eventual convergence of the aircraft to a specified trim state, controllability can significantly improve convergence speed.

3.3 Trim Database

The above procedure can be used to compute a discrete set of feasible post-failure trim conditions and then characterize each as stable, controllable, or unstable. Executing this procedure over the spectrum of flight conditions yields a multidimensional (4-D) database which, although difficult to visualize, is quite useful in planning feasible trajectories for the aircraft after a failure. This *trim database* is a discrete representation of the continuous post-failure flight envelope, completely defining the performance characteristics of the aircraft after a specific failure. Although difficult to visualize in its original form, a three-dimensional slice of the trim database can be examined by fixing one variable (like h^*) as shown in Figure 3.1. Each feasible flight condition is represented by a plotted point whose color corresponds to the two properties discussed in Section 3.2; the green points denote stable and controllable trim states, yellow denotes unstable, but controllable trim states, and the red denotes those trim states that are unstable and uncontrollable. Those trim states that are infeasible, that is, $J_{trim}^* > \epsilon_{trim}$, are not colored.

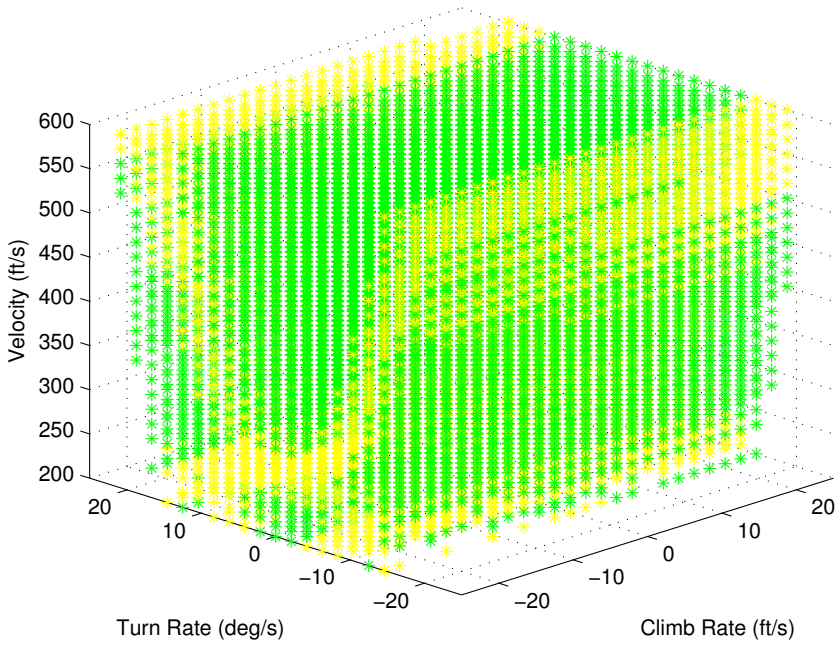


Figure 3.1: F-16 0° Rudder Failure Trim Database at Sea-Level

3.4 Trim Flight Path Displacement

When the aircraft maintains a feasible (stable or controllable) trimmed flight condition for a nonzero length of time Δt, its configuration will undergo a corresponding change. In particular, accurate flight planning is primarily concerned with identifying changes in the flight path parameters ψ and $\mathbf{p}$; The changes in θ and ϕ are implicitly defined via the specific trim states used.

3.4.1 Trim Kinematics

Aircraft flight path changes can be computed directly from the kinematic equations

$$\dot{\mathbf{p}} = \boldsymbol{R}_{\psi}^{T}\, \boldsymbol{R}_{\theta}^{T}\, \boldsymbol{R}_{\phi}^{T}\, {}^{B}\mathbf{v} \tag{3.20}$$

$$\dot{\psi} = \frac{q \sin\phi + r \cos\phi}{\cos\theta}. \tag{3.21}$$

that were presented in Chapter 2. When the aircraft is maintaining a trimmed flight condition, the change in the position and heading over the time interval Δt can be written as

$$\Delta\mathbf{p} = \int_{t_0}^{t_1} \boldsymbol{R}_{\psi}^{T}(t)\, \boldsymbol{R}_{\theta^*}^{T}\, \boldsymbol{R}_{\phi^*}^{T}\, {}^{B}\mathbf{v}^{*}(t)\, dt \tag{3.22}$$

$$\Delta\psi = \dot{\psi}^{*}\Delta t \tag{3.23}$$

where t_0 and t_1 are respectively the times at which the trim state begins and ends and $\Delta t = t_1 - t_0$. While the solution to (3.23) is intuitive, (3.22) shows that an analytic solution to the integral is possible because it consists primarily of constant trim variables.

3.4.2 Pseudo-Body Velocity

For an aircraft flying at a true trim state, the roll and pitch rotation matrices and linear velocities in the $\Delta\mathbf{p}$ equation remain constant for the duration of the flight segment. These "trim dependent" values can be grouped together to form the three dimensional vector ${}^{P}\mathbf{v}^{*}$ which describes the velocity of the trimmed aircraft with respect to an intermediate *pseudo-body* coordinate frame. The orientation of the P frame with respect to both the I and B frames is shown in Figure 3.2. The z-axis of the pseudo-body and inertial frames are aligned, while the x and y

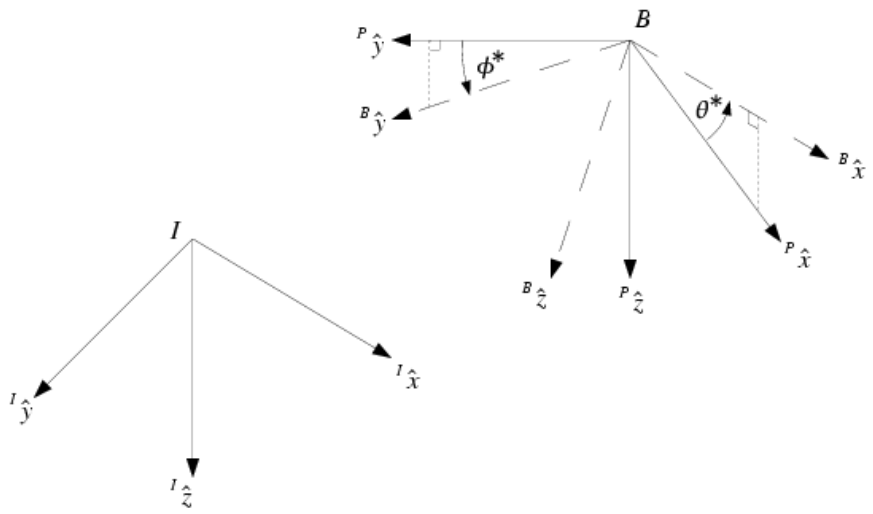


Figure 3.2: Definition of Pseudo-Body Axes Using Trim Variables

axes of the pseudo-body frame and the x and y-axis of the body-axes frame are aligned, respectively. Therefore, the P frame rotates with a change in aircraft heading while keeping its x-y plane parallel to the surface of the Earth. As a result, ${}^P\mathbf{v}^*$ remains constant during trimmed flight.

Aircraft velocity, as measured in the P frame, can be expressed as

$$ {}^P\mathbf{v}^* = \boldsymbol{R}_{\theta^*}^T \, \boldsymbol{R}_{\phi^*}^T \, {}^B\mathbf{v}^* \tag{3.24} $$

relative to the B frame trimmed velocity ${}^B\mathbf{v}^*$ and

$$ {}^P\mathbf{v}^* = \boldsymbol{R}_{\psi} \, {}^I\mathbf{v} \tag{3.25} $$

relative the inertial velocity ${}^I\mathbf{v}$. For any specific flight condition, starting at an altitude of h^*, the trimmed state variables can be determined using as shown above, which in turn can be used to compute the constant ${}^P\mathbf{v}^*$ via

$$ {}^P\mathbf{v}^* = \boldsymbol{R}_{\theta^*}^T \, \boldsymbol{R}_{\phi^*}^T \, \boldsymbol{R}_{\alpha^*}^T \, \boldsymbol{R}_{\beta^*}^T \, [v^*,\, 0,\, 0]^T. \tag{3.26} $$

3.4.3 Exact Trim Solution

Using pseudo-body coordinates, the change in position resulting from holding a constant trim state over the interval Δt can be written as

$$\Delta \mathbf{p} = \int_{t_0}^{t_1} \boldsymbol{R}_{\psi}^{T}(t)\, {}^{P}\mathbf{v}^{*}\, dt. \tag{3.27}$$

In this form, the integral can now be easily written as

$$\begin{aligned}
\Delta \mathbf{p} &= \int_{t_0}^{t_1} \boldsymbol{R}_{\psi}^{T}(t)\ {}^{P}\mathbf{v}^{*}\, dt \\
&= \boldsymbol{R}_{\psi}^{T}(t_0) \int_{0}^{\Delta t} \boldsymbol{R}_{\psi}^{T}(t)\ {}^{P}\mathbf{v}^{*}\, dt \\
&= \frac{1}{\dot{\psi}^{*}}\, \boldsymbol{R}_{\psi}^{T}(t_0) \int_{0}^{\Delta \psi} \boldsymbol{R}_{\psi}^{T}\ {}^{P}\mathbf{v}^{*}\, d\psi
\end{aligned}$$

Performing the integration, and using the explicit expression of $\boldsymbol{R}_{\psi}$ in (2.5), yields the following analytical solution:

$$\Delta \mathbf{p} = \boldsymbol{R}_{\psi_0}^{T}\, \boldsymbol{\Omega}_1\ {}^{P}\mathbf{v}^{*} \tag{3.28}$$

where

$$\boldsymbol{\Omega}_1 = \begin{cases} \dfrac{1}{\dot{\psi}^{*}} \begin{bmatrix} \sin\Delta\psi & \cos\Delta\psi - 1 & 0 \\ 1 - \cos\Delta\psi & \sin\Delta\psi & 0 \\ 0 & 0 & \Delta\psi \end{bmatrix}, & \text{if } \dot{\psi}^{*} \neq 0 \\ \Delta t \boldsymbol{I}, & \text{if } \dot{\psi}^{*} = 0 \end{cases} \tag{3.29}$$

where $\boldsymbol{R}_{\psi_0} = \boldsymbol{R}_{\psi}(t_0)$ and $\boldsymbol{I}$ is the 3×3 identity matrix.

Together, (3.23) and (3.28) fully define the flight path displacement over a trim segment held for time Δt. The new position and heading after this trim segment can be found via

$$\mathbf{p} = \mathbf{p}_0 + \boldsymbol{R}_{\psi_0}^{T}\, \boldsymbol{\Omega}_1\ {}^{P}\mathbf{v}^{*} \tag{3.30}$$

$$\psi = \psi_0 + \dot{\psi}^{*} \Delta t \tag{3.31}$$

where $\mathbf{p}_0$ and ψ_0 are the initial position and heading before the trim segment.

This result agrees with previous work of Frazzoli in mapping trimmed flight path displacements using elements of Lie algebra [28]. In fact, the screw motion Frazzoli utilizes results in helices of motion in the position and heading space is equivalent to the trimmed flight conditions defined above. To calculate the trim displacement, Frazzoli makes use of the pseudo-body velocity and coordinate frame, without explicitly defining it as such. More specifically, Frazzoli decomposes ${}^{P}\mathbf{v}^{*}$ into

$$ {}^{P}\mathbf{v}^{*} = \begin{bmatrix} V_F \cos\beta_F \cos\gamma_F \\ V_F \sin\beta_F \cos\gamma_F \\ -V_F \sin\gamma_F \end{bmatrix} \tag{3.32} $$

where $V_F = v^*$, $\gamma_F = \gamma^*$, and

$$ \tan\beta_F = \frac{\sin\beta\cos\phi + \sin\alpha\cos\beta\sin\phi}{\cos\alpha\cos\beta\cos\theta + \sin\beta\sin\phi\sin\theta + \sin\alpha\cos\beta\cos\phi\sin\theta} \tag{3.33} $$

representing the pseudo-body side slip angle. Frazzoli calls β_F the "sideslip" angle, however, it should be noted that this is different than the conventional definition presented in Section 2.2.

With this formulation, Frazzoli then defines the aircraft position displacement, with respect to the initial pseudo-body coordinate reference, as

$$ \Delta^{P}\mathbf{p} = \begin{bmatrix} r(\sin\Delta\psi\cos\beta_F + \cos\Delta\psi\sin\beta_F - \sin\beta_F) \\ r(\sin\Delta\psi\sin\beta_F - \cos\Delta\psi\cos\beta_F + \cos\beta_F) \\ V_F \sin\gamma_F \Delta t \end{bmatrix} \tag{3.34} $$

where $r = V_F \cos\gamma/\dot{\psi}^*$. Frazzoli then uses a matrix multiplication technique when computing the final inertial change in position which is equivalent to

$$ \Delta\mathbf{p} = \mathbf{p}_0 + \boldsymbol{R}^{T}_{\psi_0}\Delta^{P}\mathbf{p} \tag{3.35} $$

which immediately presents the relationship

$$\boldsymbol{\Omega}_1 \, {}^P\mathbf{v}^* = \begin{bmatrix} r(\sin\Delta\psi\cos\beta_F + \cos\Delta\psi\sin\beta_F - \sin\beta_F) \\ r(\sin\Delta\psi\sin\beta_F - \cos\Delta\psi\cos\beta_F + \cos\beta_F) \\ V_F \sin\gamma\Delta t \end{bmatrix}. \tag{3.36}$$

3.4.4 Varying Trim Solution

However, not all the trim states defined in Section 3.1.1 are constant. Slight variations in the atmosphere require the trimmed control settings $\boldsymbol{\mu}^*$, as well as the trim state $\bar{\mathbf{z}}^*$, to change in order to maintain a constant flight condition. As a result, the pseudo-body velocity also varies along climbing trim trajectories. In his research, Frazzoli assumed a uniform atmosphere and therefore neglected this variation, however, such a simplification can lead to significant errors in computing accurate ground track displacement, especially if the trim segment is held for long durations. As a fix, empirical results obtained during the current research have shown that the pseudo-body velocity varies roughly linearly with altitude, that is, $d^P\mathbf{v}/dh$ is roughly constant over a trim segment. Thus, an additional correction term can be added to (3.28) to compensate for altitude induced variations in ${}^P\mathbf{v}$.

The trim state at the beginning and end of the trimmed flight segment can be respectively computed as

$$\bar{\mathbf{z}}_i^* = \bar{\mathbf{z}}^*(h^*(t_i),\ v^*,\ \dot{h}^*,\ \dot{\psi}^*)$$

$$\bar{\mathbf{z}}_{i+1}^* = \bar{\mathbf{z}}^*(h^*(t_i) + \dot{h}^*\Delta t,\ v^*,\ \dot{h}^*,\ \dot{\psi}^*)$$

where $h^*(t_i)$ is the altitude at the beginning of the trim segment. Initial and terminal pseudo-body velocities ${}^P\mathbf{v}_i^*$ and ${}^P\mathbf{v}_{i+1}^*$ then can be computed using (3.26). The presumed linear acceleration can be expressed using the simple difference

formula

$$ {}^{P}\dot{\mathbf{v}}_i^* = \frac{{}^{P}\mathbf{v}_{i+1}^* - {}^{P}\mathbf{v}_i^*}{\Delta t} \tag{3.37} $$

where $\Delta t = t_{i+1} - t_i$. As a result, the pseudo-body velocity of the aircraft during a trim segment can be expressed as a function of time:

$$ {}^{P}\mathbf{v}^*(t) = {}^{P}\mathbf{v}_i^* + {}^{P}\dot{\mathbf{v}}_i^*(t - t_i). \tag{3.38} $$

By substituting (3.38) for (3.26), (3.27) can now be written as

$$ \Delta\mathbf{p} = \int_{t_0}^{t_1} \boldsymbol{R}_\psi^T(t)\big({}^{P}\mathbf{v}_i^* + {}^{P}\dot{\mathbf{v}}_i^*(t - t_0)\big)\, dt. \tag{3.39} $$

Using similar simplification and variable substitution steps from Section 3.4.3 yields

$$ \begin{aligned} \Delta\mathbf{p} &= \int_{t_0}^{t_1} \boldsymbol{R}_\psi^T(t)\big({}^{P}\mathbf{v}_i^* + {}^{P}\dot{\mathbf{v}}_i^*(t - t_0)\big)\, dt. \\ &= \boldsymbol{R}_\psi^T(t_0) \int_0^{\Delta t} \boldsymbol{R}_\psi^T(t)\big({}^{P}\mathbf{v}_i^* + {}^{P}\dot{\mathbf{v}}_i^*\, t\big)\, dt. \\ &= \boldsymbol{R}_\psi^T(t_0)\left[\int_0^{\Delta t} \boldsymbol{R}_\psi^T(t)\, {}^{P}\mathbf{v}_i^*\, dt + \int_0^{\Delta t} \boldsymbol{R}_\psi^T(t)\, {}^{P}\dot{\mathbf{v}}_i^*\, t\, dt\right] \\ &= \boldsymbol{R}_\psi^T(t_0)\left[\frac{1}{\dot{\psi}^*}\int_0^{\Delta\psi} \boldsymbol{R}_\psi^T\, {}^{P}\mathbf{v}_i^*\, d\psi + \frac{1}{\dot{\psi}^{*2}}\int_0^{\Delta\psi} \boldsymbol{R}_\psi^T\, {}^{P}\dot{\mathbf{v}}_i^*\, \psi\, d\psi\right] \end{aligned} $$

which can be solved analytically as

$$ \Delta\mathbf{p} = \boldsymbol{R}_{\psi_0}^T\big(\boldsymbol{\Omega}_1\, {}^{P}\mathbf{v}_i^* + \boldsymbol{\Omega}_2\, {}^{P}\dot{\mathbf{v}}_i^*\big) \tag{3.40} $$

with

$$ \boldsymbol{\Omega}_2 = \begin{cases} \dfrac{1}{\dot{\psi}^{*2}} \begin{bmatrix} \cos\Delta\psi + \Delta\psi\sin\Delta\psi - 1 & -\sin\Delta\psi + \Delta\psi\cos\Delta\psi & 0 \\ \sin\Delta\psi - \Delta\psi\cos\Delta\psi & \cos\Delta\psi + \Delta\psi\sin\Delta\psi - 1 & 0 \\ 0 & 0 & \Delta\psi^2 \end{bmatrix}, & \text{if } \dot{\psi}^* \neq 0 \\ \dfrac{\Delta t^2}{2}\boldsymbol{I}, & \text{if } \dot{\psi}^* = 0 \end{cases} \tag{3.41} $$

It is important to note that the empirical linear relationship between the ${}^{P}\mathbf{v}^{*}$ and altitude is not precise. However, the method of incorporating an additional correction term for this variation is general to any type of pseudo-body velocity/altitude relationship. In fact, (3.38) is a first-order Taylor series expansion of the true ${}^{P}\mathbf{v}(t)$, and it is possible to improve accuracy, as well as account for more general pseudo-body velocity shapes, with each additional term in the expansion. For this work, the additional term $\boldsymbol{\Omega}_2 \, {}^{P}\dot{\mathbf{v}}_i^{*}$ is necessary for accurate calculation of (3.27) for rudder and aileron failure trim states. The severity of these failures will be shown to require side slipping flight that results in larger, though still linear, variations in the pseudo-body velocity. For example, over a typical 10,000 ft flight path, Frazzoli's uniform trim displacement kinematics leads to ground track errors of $\sim$10 ft, whereas, the new higher-order expansion (3.40) reduces this error to about 1 inch.

3.5 Aircraft Kinematic Model

Using the analysis above, a simplified kinematic model is now described that maps sequences of trimmed flight conditions to flight path displacements. This model is analogous to the method of mapping joint space to Cartesian space used in robotics, and, as a result, will use much of the same notation.

A useful, compact representation of the aircraft's flight path configuration uses the 4×4 matrix [35]

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{R}_{\psi}^{T} & \mathbf{p} \\ 0 & 1 \end{bmatrix}. \tag{3.42}$$

The effect of holding a single trim state for a time Δt, shown above in (3.40),

can be similarly represented using the mapping [28]

$$\boldsymbol{G}(v^*,\ \dot{h}^*,\ \dot{\psi}^*, \Delta t) = \begin{bmatrix} \boldsymbol{R}_{\Delta\psi}^T & \boldsymbol{\Omega}_1{}^P\mathbf{v}_i^* + \boldsymbol{\Omega}_2{}^P\dot{\mathbf{v}}_i^* \\ 0 & 1 \end{bmatrix} \tag{3.43}$$

where

$$\boldsymbol{R}_{\Delta\psi} = \begin{bmatrix} \cos(\dot{\psi}^*\Delta t) & \sin(\dot{\psi}^*\Delta t) & 0 \\ -\sin(\dot{\psi}^*\Delta t) & \cos(\dot{\psi}^*\Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.44}$$

The specific values ${}^P\mathbf{v}^*(t_0)$ and ${}^P\dot{\mathbf{v}}^*$ can be determined by using the altitude specified in $\boldsymbol{F}$. After this maneuver, the new flight path configuration can be computed as [28]

$$\boldsymbol{F}_{new} = \boldsymbol{F}\boldsymbol{G}(v^*,\ \dot{h}^*,\ \dot{\psi}^*, \Delta t). \tag{3.45}$$

More generally, the total change in position and heading after a sequence of N consecutive trim states, held for durations Δt_i, can be represented by

$$\boldsymbol{F}_N = \boldsymbol{F}_0 \prod_{i=1}^{N} \boldsymbol{G}_{i-1,i}\boldsymbol{G}(v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*, \Delta t_i) \tag{3.46}$$

where $\boldsymbol{F}_0$ is the initial aircraft flight path configuration and $\boldsymbol{G}_{i-1,i}$ describes the flight path displacement over the transition from trim state $i-1$ to i. Note that the $\boldsymbol{G}_{0,1}$ transition allows the initial trim state at $\boldsymbol{F}_0$ to differ from the trim state used in $\boldsymbol{G}(v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*, \Delta t_i)$.

3.5.1 Kinematic Databases

When using (3.46), computing the trim states and their associated transitions for each $\boldsymbol{F}_N$ calculation would require more time and computational resources than available. A more practical approach is to precompute a representative set and store these values in database form. The database can be accessed via

an interpolation routine to approximate intermediate values. The trim database already contains trim state information and therefore can be used to compute the database T containing pseudo-body velocity values for a variety of flight conditions. The input/output operation of T can be expressed by the function

$$^{P}\mathbf{v}^{*} = T(h^{*},\ v^{*},\ \dot{h}^{*},\ \dot{\psi}^{*}). \tag{3.47}$$

Similarly, a database M can be computed containing values for $\boldsymbol{G}_{i-1,i}$ associated with transitions between any two trimmed flight conditions found in T. Since $\boldsymbol{G}_{i-1,i}$ also varies with density, the input/output operation of M can be expressed by the function

$$\boldsymbol{G}_{i-1,i} = M(h^{*},\ v^{*}_{i-1},\ \dot{h}^{*}_{i-1},\ \dot{\psi}^{*}_{i},\ v^{*}_{i},\ \dot{h}^{*}_{i},\ \dot{\psi}^{*}_{i}). \tag{3.48}$$

Unlike the configuration change during a trim segment, the computation of individual $\boldsymbol{G}_{i-1,i}$ cannot be performed using the trim states presented above. In fact, computing $\boldsymbol{G}_{i-1,i}$ requires a full dynamic analysis of the aircraft. Such an analysis, and explicit definition of $\boldsymbol{G}_{i-1,i}$ is presented in the next chapter.

3.6 Trim Controller

Using the method from Section 3.1.2, an open-loop trim controller C capable of commanding all feasible trim trajectories can be defined as

$$C(h^{*},\ v^{*},\ \dot{h}^{*},\ \dot{\psi}^{*}) = \boldsymbol{\mu}^{*} \tag{3.49}$$

which translates the current altitude and desired flight condition into the required trimmed control settings. Such a controller is shown coupled with an aircraft dynamic simulator AC in Figure 3.3. To compensate for the constant adjustment

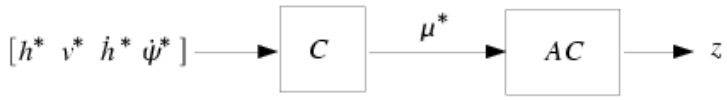


Figure 3.3: Trim Controller Block Diagram

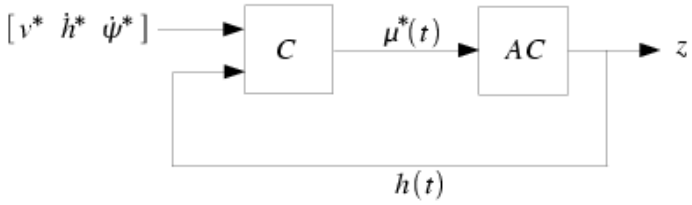


Figure 3.4: Trim Controller Block Diagram with Altitude Feedback

to $\boldsymbol{\mu}^*$ during climbing flight, the altitude loop is closed as shown in Figure 3.4. While relatively trivial in its current form, provides a foundation on which more sophisticated closed-loop controllers can be built.

3.6.1 Controller Scheduling

In practice, the method of computing the required trimmed control settings is not accomplished by performing the actual calculations discussed in Section 3.1.2 in the control loop. The computational complexity of the constrained minimization would result in extremely slow controller update rates. Instead, C contains the trimmed control settings from the trim database and can be scheduled according to the actual flight condition supplied on-line using a four-dimensional interpolation routine.

Chapter 4

Transition Analysis

The emergency flight planner builds post-failure flight trajectories from a sequence of trimmed flight conditions. Trim states have been defined with an aircraft kinematic analysis. This chapter presents the dynamic and control analysis required to accurately characterize the transition between trim states. A closed-loop control law is defined that reduces transition settling time and that provides close tracking of a desired flight path.

Section 4.1 defines trim transitions and demonstrates that open-loop control techniques are unable to produce adequate transition performance. Section 4.2 examines the reasons the open-loop system failed and and defines a full-state feedback control law to correct these deficiencies. Section 4.3 adds an integral term to the closed-loop controller to provide configuration tracking capabilities. Finally, Section 4.4 provides an explicit definition for $\boldsymbol{G}_{i-1,i}$ and introduces the concept of transition connectivity.

4.1 Open-Loop Transitions

4.1.1 Trim Transitions

A trim transition is defined as a finite time change between two trim states. For an aircraft traveling along a trim trajectory specified by the constant initial flight condition v_i^*, $\dot{h}_i^*$, and $\dot{\psi}_i^*$, a trim transition corresponds to the change

$$(v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*) \rightarrow (v_j^*,\ \dot{h}_j^*,\ \dot{\psi}_j^*) \tag{4.1}$$

over time Δt, where $(v_j^*,\ \dot{h}_j^*,\ \dot{\psi}_j^*)$ defines the terminal flight condition. The results of the trim analysis in Chapter 3 show that (4.1) requires the corresponding change in the trim state variables

$$\bar{\mathbf{z}}_i^*(h_i,\ v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*) \rightarrow \bar{\mathbf{z}}_j^*(h_j,\ v_j^*,\ \dot{h}_j^*,\ \dot{\psi}_j^*) \tag{4.2}$$

where h_i and h_j are respectively the initial and resulting terminal altitude. Similarly, the control input must also be changed so that

$$\boldsymbol{\mu}_i^*(h_i,\ v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*) \rightarrow \boldsymbol{\mu}_j^*(h_j,\ v_j^*,\ \dot{h}_j^*,\ \dot{\psi}_j^*) \tag{4.3}$$

where $\boldsymbol{\mu}_i^*$ and $\boldsymbol{\mu}_j^*$ are respectively the trimmed control settings for the trim states $\bar{\mathbf{z}}_i^*$ and $\bar{\mathbf{z}}_j^*$.

When performing the above transitions, there are many methods to quantify the configuration and control setting changes over the maneuver. Generally, there are three different strategies a transition can be performed open-loop, i.e., without the aid of state feedback. Moreover, when evaluating these open-loop methods, the performance of the nonlinear response must be considered. Normally, good performance is characterized by quick rise and settling times and little overshoot.

4.1.2 Discontinuous Transitions

The simplest method to characterize (4.1) is instantaneously switching actuator settings from $\boldsymbol{\mu}_i^*$ to $\boldsymbol{\mu}_j^*$, where a value of $h_j = h_i$ can be used to determine the specific value of the terminal trimmed control setting. To see the performance of such a discontinuous control law, Figure 4.1 shows the flight condition response for an example transition from

$$v_i^* = 400 \text{ ft/s}, \; \dot{h}_i^* = -8.33 \text{ ft/s}, \; \dot{\psi}_i^* = -3 \text{ deg/s}$$

to

$$v_j^* = 425 \text{ ft/s}, \; \dot{h}_j^* = 0 \text{ ft/s}, \; \dot{\psi}_j^* = 0 \text{ deg/s}$$

using the nominal F-16 aircraft model at an initial altitude of $h_i = 1000$ ft the control signal shown in Figure 4.2. While this discontinuous method may be the easiest open-loop control law to implement, its performance characteristics are poor, resulting in large short-term transients and long settling times.

4.1.3 Trim Interpolated Transitions

A more logical approach would be smoothly varying the actuator settings over time Δt. With this strategy, the actuator variation, initiating at time t_0, could be linearly transitioned by

$$\boldsymbol{\mu}^*(t) = \boldsymbol{\mu}_i^* + \frac{\boldsymbol{\mu}_j^* - \boldsymbol{\mu}_i^*}{\Delta t}(t - t_0) \tag{4.4}$$

for all $t \in [t_0, \; t_0 + \Delta t]$ where the terminal control setting $\boldsymbol{\mu}_j^*$ can be found from the terminal altitude estimate

$$h_j = h_i + \frac{\dot{h}_j^* + \dot{h}_i^*}{2}\Delta t.$$

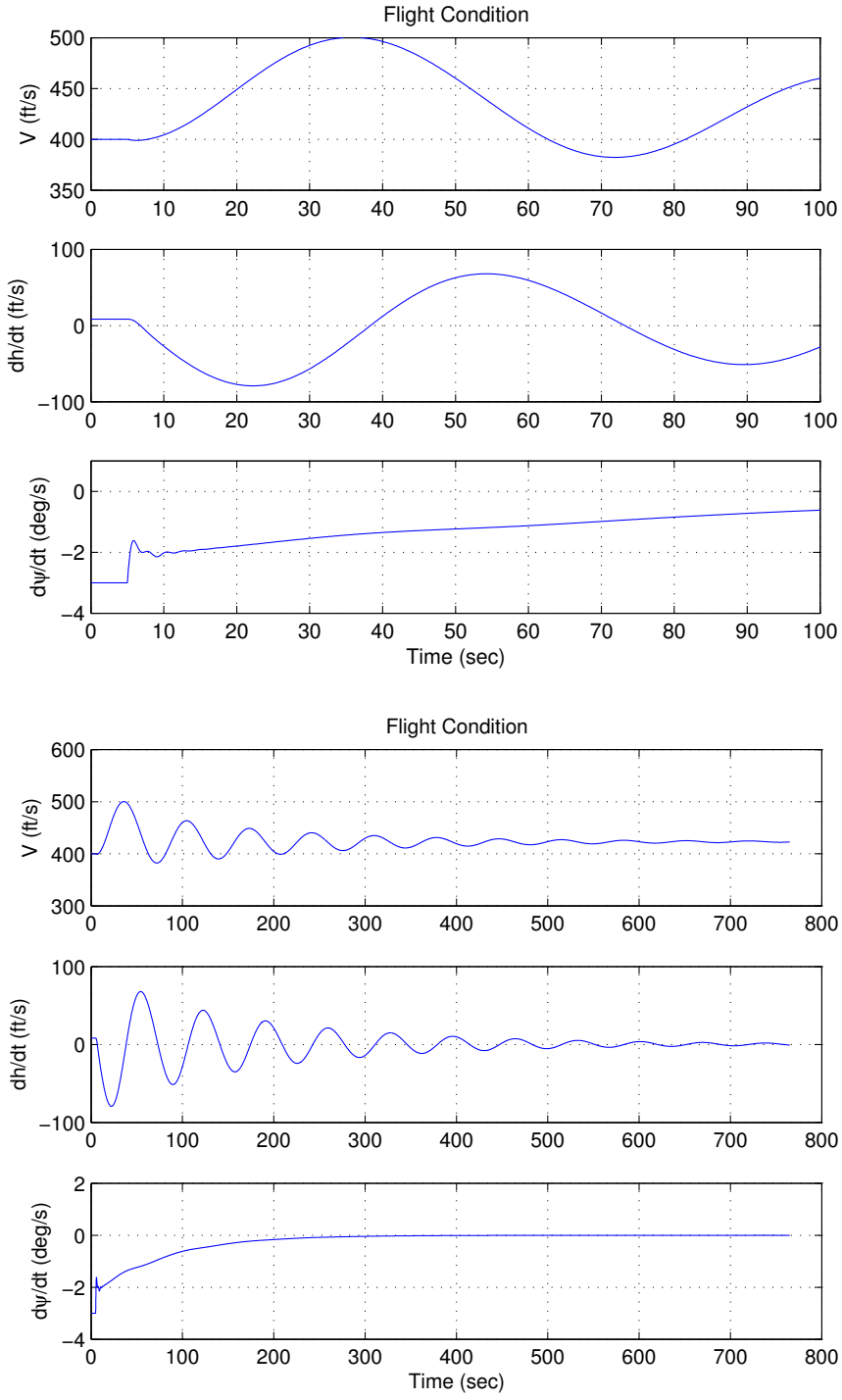


Figure 4.1: Flight Condition Response using Discontinuous Controller

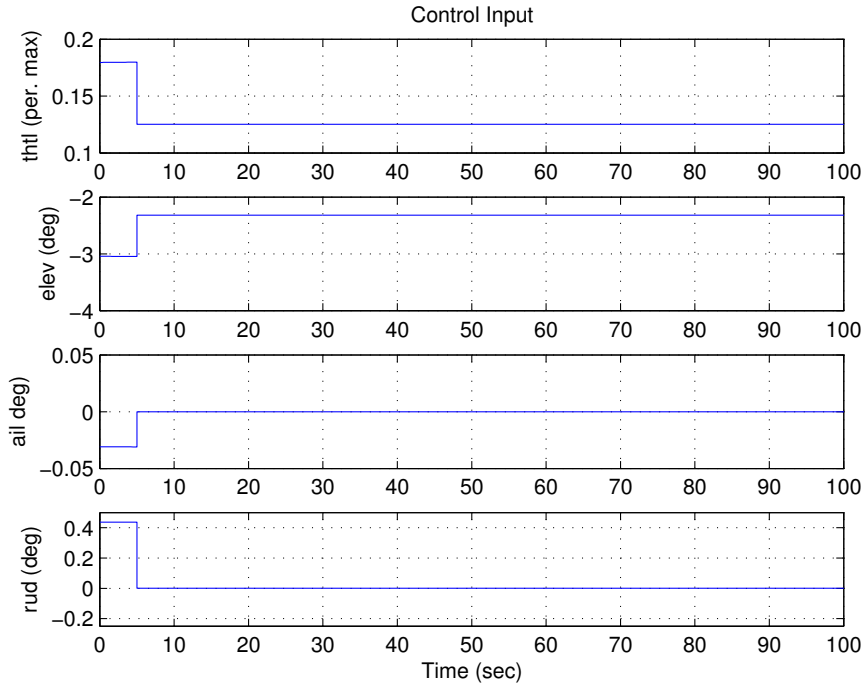


Figure 4.2: Applied Control Input using Discontinuous Controller

The performance of this controller can be gauged by observing the aircraft response in Figure 4.3 which shows the example transition defined above being commanded with the applied control signal, provided in Figure 4.4. Again, the presence of large transients and long settling times shows poor controller performance. The applied control signal in this example used a $\Delta t = 20$ sec. While the magnitude of the transient can be reduced by increasing Δt, this further increases the time it takes the aircraft to arrive at the terminal flight condition.

4.1.4 Trajectory Interpolated Transition

As an alternative to directly varying the actuator settings in an attempt to smooth the transition, the interpolation can instead be performed on flight condition variation. This is done by defining an *ideal* transition explicitly describing the desired flight condition variation during the maneuver as

$$\begin{aligned} v^*(t) &= v_i^* + \frac{v_j^* - v_i^*}{\Delta t}(t - t_0) \\ \dot{h}^*(t) &= \dot{h}_i^* + \frac{\dot{h}_j^* - \dot{h}_i^*}{\Delta t}(t - t_0) \\ \dot{\psi}^*(t) &= \dot{\psi}_i^* + \frac{\dot{\psi}_j^* - \dot{\psi}_i^*}{\Delta t}(t - t_0). \end{aligned} \tag{4.5}$$

The altitude-feedback controller C from Section 3.6 can be used to translate (4.5), or any other flight condition interpolation, into the equivalent changes in trimmed actuator settings. For comparison with the other open-loop techniques, the flight condition response using this technique is shown Figure 4.5 and the actuator signal supplied by C is displayed in Figure 4.6.

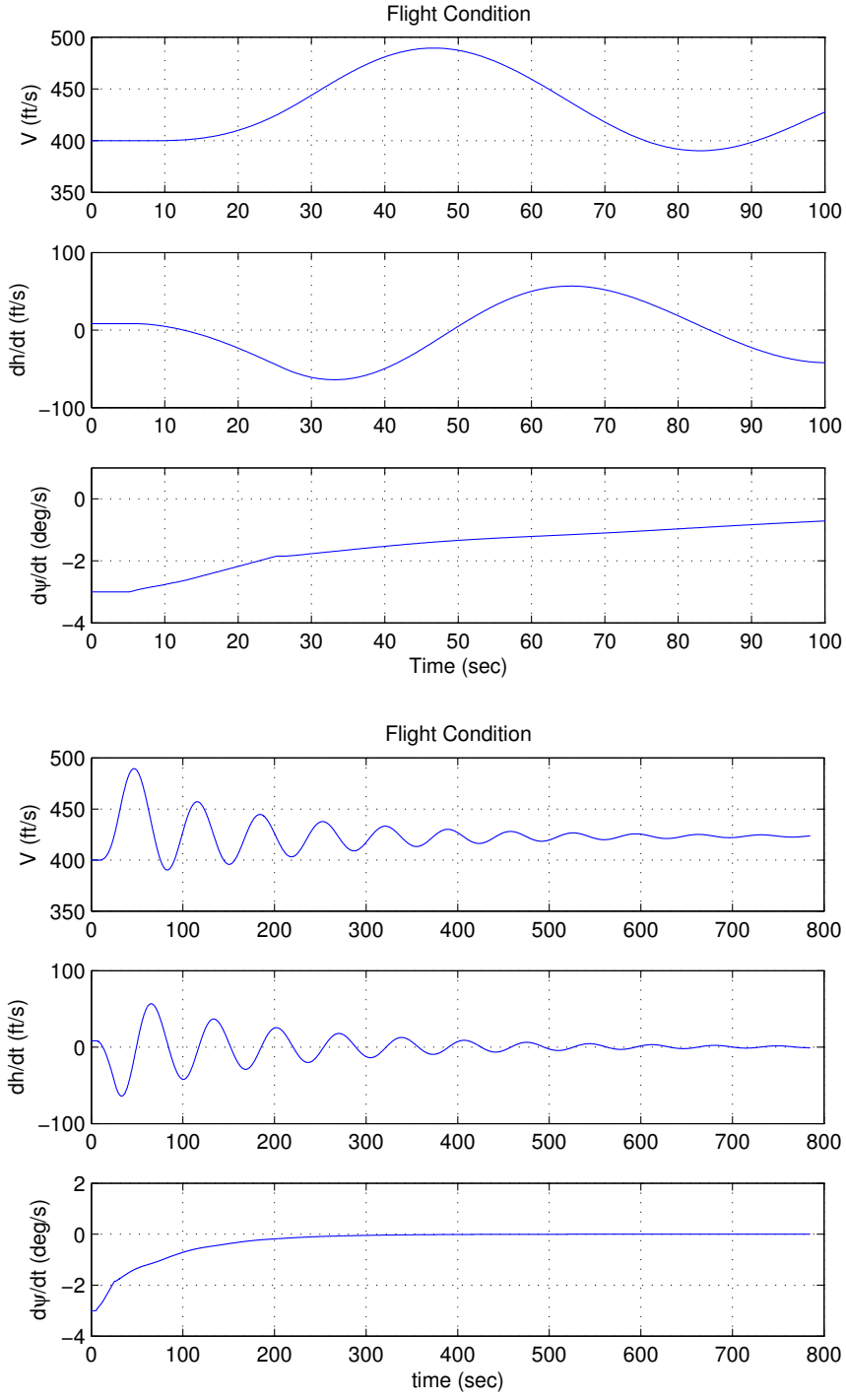


Figure 4.3: Flight Condition Response using Trim Interpolated Controller

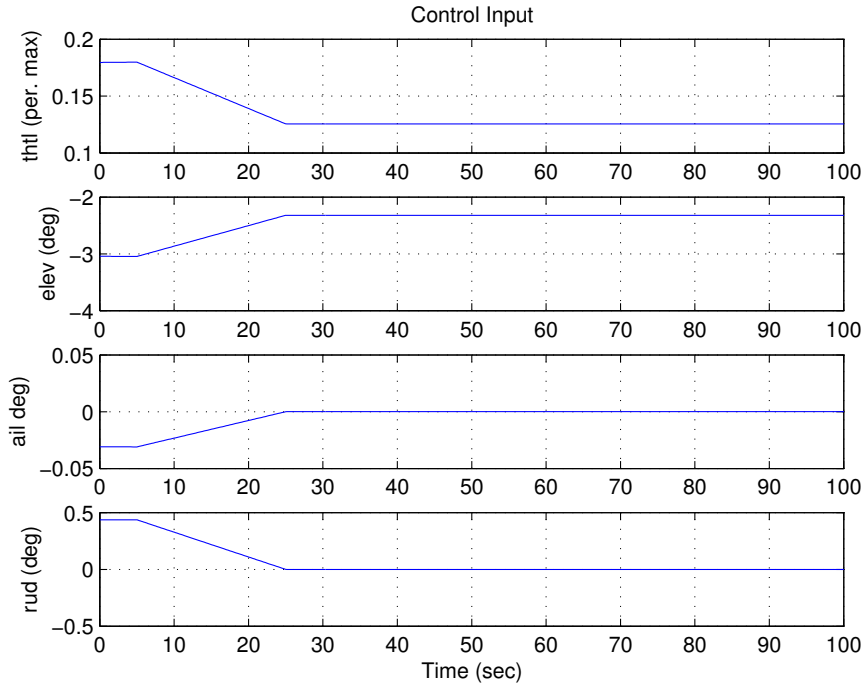


Figure 4.4: Applied Control Input using Trim Interpolated Controller

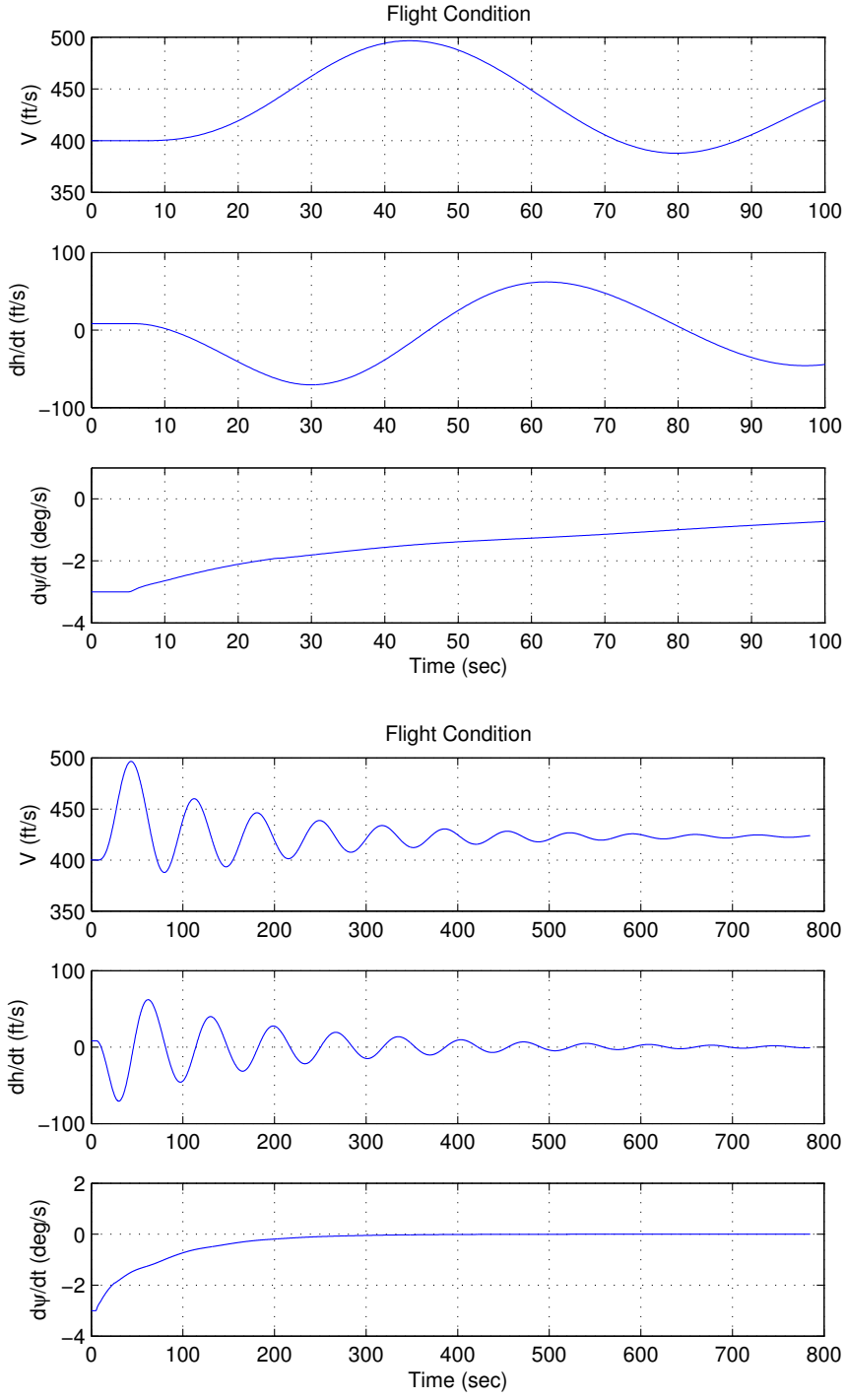


Figure 4.5: Flight Condition Response using Trajectory Interpolated Controller

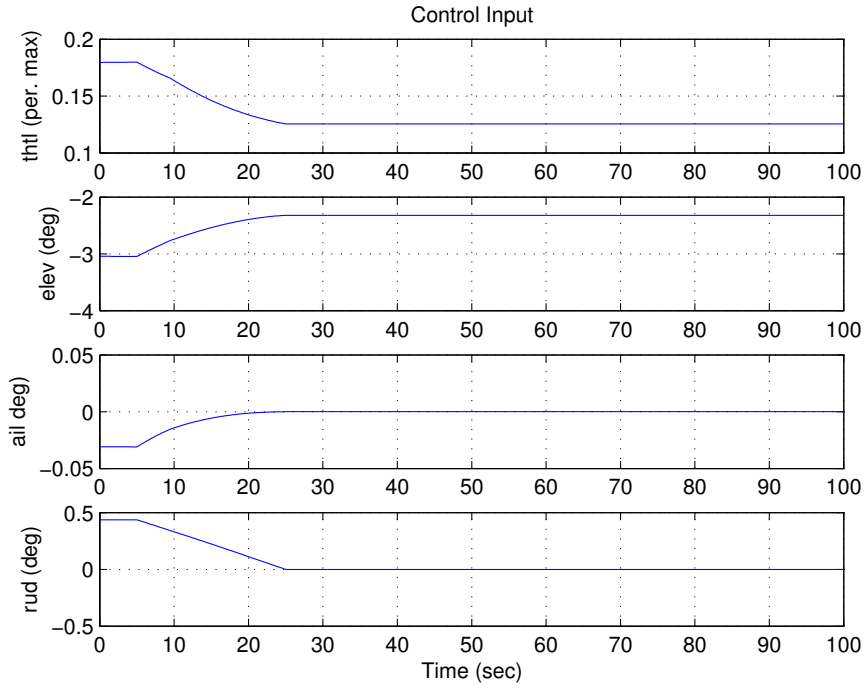


Figure 4.6: Applied Control Input using Trajectory Interpolated Controller

4.2 Performance Tracking

Despite smooth actuation strategies, open-loop controllers provide poor transition performance because they unable to compensate for the natural aircraft dynamics that create large amplitude and long duration transients. To combat these transients, a more active feedback control strategy is required. As will be shown below, each additional layer of feedback control design improves the quality of the transition by reducing the undesirable transients and by partially tracking an ideal flight path response. The added tracking in h and ψ minimizes possible deviations to promote accurate mapping of the overall transition flight path displacement in the simplified kinematic aircraft model.

4.2.1 Transient Behavior

To understand how to design a compensating controller, it is first necessary to examine sources of poor open-loop performance. Recall that, in small neighborhoods surrounding the trim states $\bar{\mathbf{z}}_k^*$, the nonlinear aircraft dynamics can be approximated by (3.11) (see Section 3.2). This simplification provides a more straightforward method of determining transient performance characteristics than complex nonlinear stability techniques such as Lyapunov analysis [32]. As will be shown below, the performance of a linearized dynamic system can be determined by examining the eigenvalues of the Jacobian matrix $\boldsymbol{A}_k$.

Good performance is typically characterized by quick rise and settling times, as well as minimal oscillations. While specific nonlinear system responses depend on both $\boldsymbol{A}_k$'s eigenvectors and eigenvalues, as a general rule, the time domain criteria associated with good performance translates to requiring all the eigenvalues of the system to lie inside the region Λ, shown in Figure 4.7 [33], which is

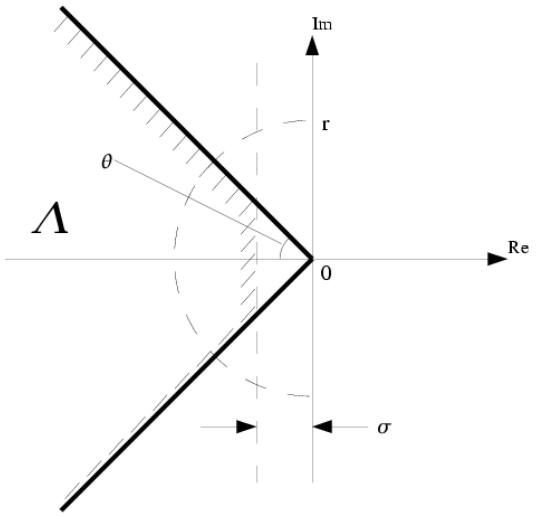


Figure 4.7: Performance Minded Eigenvalue Location

bounded on the right by a vertical line, and from above and below by two lines emanating from the origin. Increasing σ decreases settling time and decreasing θ reduces transient oscillation magnitude.

However, typical aircraft performance yields eigenvalues outside Λ. The presence of the phugoid mode, a slow, lightly damped oscillatory mode coupling altitude and velocity, and the spiral mode, a slow exponential mode coupling the roll angle and yaw rate, result in poor open-loop transition performance. The contribution of these natural modes can be seen in Figures 4.1, 4.3, and 4.5. The phugoid mode drives the velocity and climb rate oscillations, exchanging potential and kinetic energy within the system as the aircraft slowly settles to an energy equilibrium. The spiral mode causes slow convergence of $\dot{\psi}$. Thus, providing better transition performance requires a controller that can alter the location of the eigenvalues of the closed-loop system.

4.2.2 Nonlinear PD Control

As previously discussed, if the linearized aircraft dynamics are controllable, then the state feedback control law

$$\mathbf{u}_k - \boldsymbol{K}_k \mathbf{x}_k$$

is capable of arbitrarily placing the resulting closed-loop eigenvalues. More specifically, it was shown that if

$$\dot{\mathbf{x}}_k = \boldsymbol{A}_k \mathbf{x}_k + \boldsymbol{B}_k \mathbf{u}_k$$

was controllable, then there exists a specific choice for $\boldsymbol{K}_k$ such that the closed-loop system

$$\dot{\mathbf{x}}_k = (\boldsymbol{A}_k - \boldsymbol{B}_k \boldsymbol{K}_k)\mathbf{x}_k$$

has the property

$$\lambda_i(\boldsymbol{A}_k - \boldsymbol{B}_k \boldsymbol{K}_k) = \lambda_{des,i} \quad \forall i = 1, \ldots, 8.$$

Thus, aircraft controllability, at least locally around trim states, is necessary to reshape the transition transients. The open-loop examples of Section 4.1 show that stability alone is insufficient when performing transitions and that controllability should be the driving factor in choosing what trim states are suitable for the trajectory flight path planner. This freedom to reshape the transients is what drives the desire to grade feasible trim states according to their controllability.

The above analysis can be used to design a full-state feedback nonlinear controller. The trajectory interpolated open-loop controller, discussed in Section 4.1.4, can be augmented with the feedback control signal

$$\mathbf{u}(t) = -\boldsymbol{K}(t)\mathbf{x}(t),$$

where

$$\mathbf{x}(t) = \bar{\mathbf{z}}(t) - \bar{\mathbf{z}}^*(t),$$

to form the nonlinear control law

$$\boldsymbol{\mu}(h(t),\, v^*(t),\, \dot{h}^*(t),\, \dot{\psi}^*(t)) = \boldsymbol{\mu}^*(t) - \boldsymbol{K}(t)\big(\bar{\mathbf{z}}(t) - \bar{\mathbf{z}}^*(t)\big). \tag{4.6}$$

where $h(t)$ represents altitude-feedback and $v^*(t)$, $\dot{h}^*(t)$, and $\dot{\psi}^*(t)$ are varied according to (4.5). At each instant in time, the specified flight condition is used to to compute the instantaneous feed-forward trim control setting $\boldsymbol{\mu}^*(t)$, which effectively linearizes the aircraft at that flight condition. The controller also computes the corresponding trim state $\bar{\mathbf{z}}^*(t)$ which acts as the desired tracking response and the feedback gain matrix $\boldsymbol{K}(t)$ which places the closed-loop eigenvalues of the instantaneous linear system at a specified $\boldsymbol{\lambda}_{des}$.

As mentioned above, (4.6) is commonly referred to as a state feedback control law, but it also loosely resembles a classical proportional-plus-derivative (PD) controller. Traditionally, a PD controller produces feedback signal $\mathbf{u}(t)$ that changes proportional to a change in an error signal $\mathbf{e}(t)$ and to the rate at which the error signal changes ($d\mathbf{e}(t)/dt$). In this case, no such error signal is present, however, the deviation $\mathbf{x}(t)$ acts as a combined measure of the error in the trim state, through deviations in $\theta(t)$ and $\phi(t)$, as well as the rate at which the trim state error is evolving, through deviations in $\boldsymbol{\nu}(t)$. An input/output block diagram of (4.6) is provided in Figure 4.8.

4.2.3 Closed-Loop Eigenvalue Placement

Although aircraft controllability allows for the general placement of the closed-loop eigenvalues of the linearized aircraft dynamics, in practice, additional concerns arise in the selection of reasonable values when designing $\boldsymbol{K}_k$. If the desired

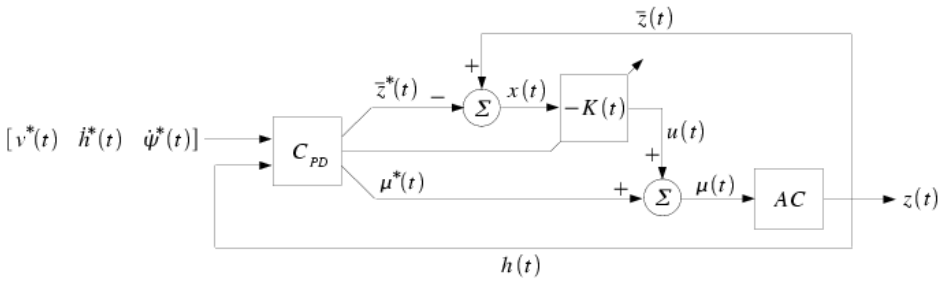


Figure 4.8: PD Control Block Diagram

eigenvalues are closely-spaced, the time response of the closed-loop system will tend to be slow and the feedback control signal will be large. A straightforward method of selecting appropriate $\boldsymbol{\lambda}_{des}$ is to place them around the circle of radius r within region Λ as shown in Figure 4.7 [33]. Increasing r decreases transient settling times but requires larger actuation signals. While this work adopted this straightforward eigenvalue placement technique, additional methods do exist in optimal control literature able to explicitly handle trade-offs between transient settling times and actuator power [36–38].

4.2.4 State Feedback Scheduling

In practice, C_{PD} is scheduled, in much the same way as C in Section 3.6. Specifically, values for $\boldsymbol{\mu}_k^*$, $\bar{\mathbf{z}}_k^*$, and $\boldsymbol{K}_k$ would be computed at a variety of flight conditions $(h_k^*,\ v_k^*,\ \dot{h}_k^*,\ \dot{\psi}_k^*)$ and stored in a tabular database. On-line, the controller would then supply approximate values for $\boldsymbol{\mu}^*(t)$, $\bar{\mathbf{z}}^*(t)$, and $\boldsymbol{K}(t)$ by performing a four-dimensional table lookup on the database using the current specified flight condition $(h^*(t),\ v^*(t),\ \dot{h}^*(t),\ \dot{\psi}^*(t))$. This scheduled controller was used to produce the response, provided in Figure 4.9. For reference, the ideal flight condition response and trim control settings are also provided. The benefits of utilizing the

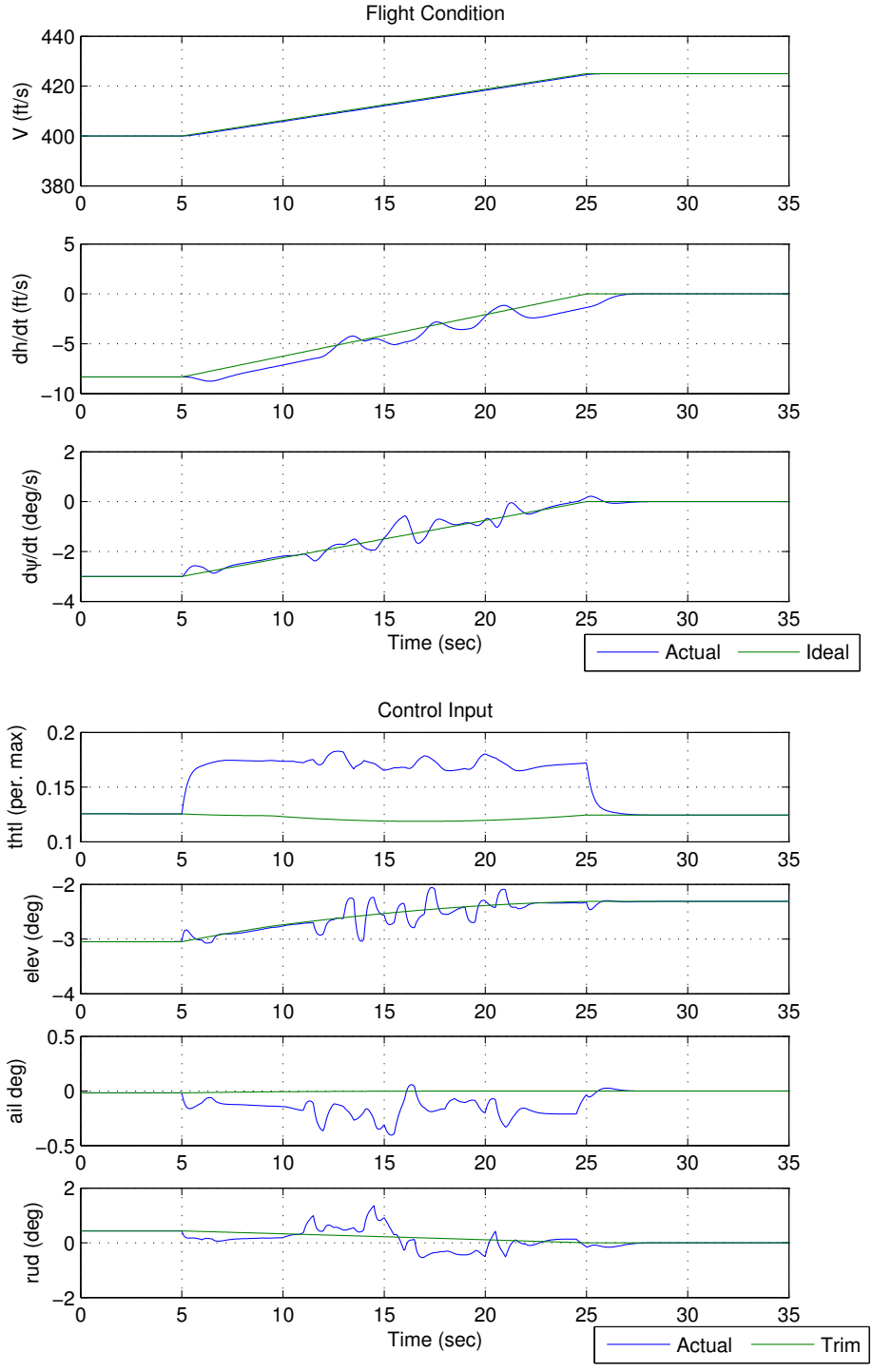


Figure 4.9: PD Controller Response

full-state feedback controller are immediately apparent. The transient settling time and oscillation magnitudes have been drastically reduced. Note that the jagged response is a direct result of scheduling, i.e., a lack of smoothness in the supplied $\bar{\mathbf{z}}^*(t)$, $\boldsymbol{\mu}^*(t)$, and $\boldsymbol{K}(t)$.

4.3 Configuration Tracking

Besides good performance characteristics, trim transitions must also produce repeatable flight paths for the simplified aircraft kinematic model to accurately predict vehicle motion. Small deviations in the measured position or heading after a transition can potentially produce even larger errors after the succeeding trim segment. The influence of these errors can be limited by designing a control law capable of tracking desired flight paths specified by the ideal transition—(4.5) implicitly defines an ideal $\mathbf{p}^*(t)$, and $\psi^*(t)$ through the kinematic effect of $\boldsymbol{\nu}^*(t)$. Figure 4.10 shows that the full-state feedback controller of the previous section cannot provide the desired tracking objective. However, augmenting a full-state feedback control with an additional control loop coupling corrective actuator deflection rates flight condition deviations can provide better tracking accuracy. To guarantee closed-loop stability, though, both the integral and full-state feedback controllers must be designed simultaneously.

4.3.1 MIMO Linearized Dynamics

Analyzing this coupled stability requires viewing the linearized aircraft dynamics as the multi-input multi-output system

$$\begin{aligned} \dot{\mathbf{x}}_k &= \boldsymbol{A}_k \mathbf{x}_k + \boldsymbol{B}_k \mathbf{u}_k \\ \mathbf{y}_k &= \boldsymbol{C}_k \mathbf{x}_k \end{aligned} \tag{4.7}$$

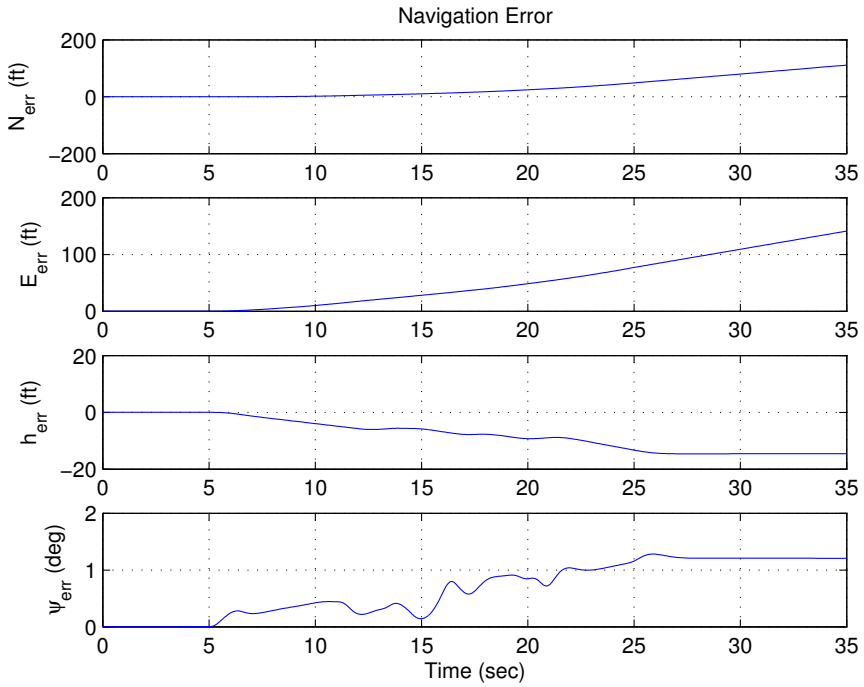


Figure 4.10: Ground Track Error using PD Controller

where now $\mathbf{y}_k$ is the output measurement of flight condition perturbations

$$\mathbf{y}_k = \begin{bmatrix} v - v_k^* \\ \dot{h} - \dot{h}_k^* \\ \dot{\psi} - \dot{\psi}_k^* \end{bmatrix} \tag{4.8}$$

and the matrix $\mathbf{C}_k$ is the constant matrix

$$\mathbf{C}_k = \frac{\partial \mathbf{y}_k}{\partial \mathbf{z}} |_{\mathbf{z}_k^*, \boldsymbol{\mu}_k^*}. \tag{4.9}$$

Unlike $\boldsymbol{A}_k$ and $\mathbf{B}_k$, (4.9) can be computed analytically using the kinematic relationships for the climb and turn rates

$$\dot{h} = \cos\alpha \cos\beta \sin\theta - \sin\phi \sin\beta \cos\theta - \cos\phi \sin\alpha \cos\beta \cos\theta$$

$$\dot{\psi} = \frac{q \sin\phi + r \cos\phi}{\cos\theta}$$

and evaluating the partial derivatives at the trimmed flight condition. As a result, $\mathbf{C}_k$ can be expressed, component-wise, as

$$c_{1,1} = 1$$

$$c_{2,1} = \cos\alpha_k^* \cos\beta_k^* \sin\theta_k^* - \sin\phi_k^* \sin\beta_k^* \cos\theta_k^* - \cos\phi_k^* \sin\alpha_k^* \cos\beta_k^* \cos\theta_k^*$$

$$c_{2,2} = v_k^*(-\sin\alpha_k^* \cos\beta_k^* \sin\theta_k^* - \cos\phi_k^* \cos\alpha_k^* \cos\beta_k^* \cos\theta_k^*)$$

$$c_{2,3} = v_k^*(\cos\alpha_k^* \cos\beta_k^* \cos\theta_k^* + \sin\phi_k^* \sin\beta_k^* \sin\theta_k^* + \cos\phi_k^* \sin\alpha_k^* \cos\beta_k^* \sin\theta_k^*)$$

$$c_{2,5} = v_k^*(-\cos\alpha_k^* \sin\beta_k^* \sin\theta_k^* - \sin\phi_k^* \cos\beta_k^* \cos\theta_k^* + \cos\phi_k^* \sin\alpha_k^* \sin\beta_k^* \cos\theta_k^*)$$

$$c_{2,6} = v_k^*(-\cos\phi_k^* \sin\beta_k^* \cos\theta_k^* + \sin\phi_k^* \sin\alpha_k^* \cos\beta_k^* \cos\theta_k^*)$$

$$c_{3,3} = \frac{(q_k^* \sin\phi_k^* + r_k^* \cos\phi_k^*) \sin\theta_k^*}{\cos\theta_k^{*2}}$$

$$c_{3,4} = \frac{\sin\phi_k^*}{\cos\theta_k^*}$$

$$c_{3,6} = \frac{q_k^* \cos\phi_k^* + r_k^* \sin\phi_k^*}{\cos\theta_k^*}$$

$$c_{3,8} = \frac{\cos\phi_k^*}{\cos\theta_k^*}$$

where $c_{i,j}$ is the component of $\mathbf{C}_k$ located in the i^{th} row and j^{th} column. All components not listed are zero.

As a result, the integral feedback control law then can be expressed as

$$\dot{\mathbf{u}}_k = -\boldsymbol{K}_{i,k}\mathbf{y}_k, \tag{4.10}$$

where $\mathbf{K}_{i,k}$ is a constant output feedback gain matrix. Coupling controller and linearized aircraft dynamics yields the augmented system

$$\begin{bmatrix} \dot{\mathbf{x}}_k \\ \dot{\boldsymbol{\xi}}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_k & \mathbf{0} \\ -\boldsymbol{C}_k & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\xi}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_k \\ \mathbf{0} \end{bmatrix} \mathbf{u}_k \tag{4.11}$$

where $\dot{\boldsymbol{\xi}}_k = -\mathbf{y}_k$. The substitution

$$\boldsymbol{\zeta}_k^T = [\mathbf{x}_k^T \ \ \boldsymbol{\xi}_k^T]$$

allows (4.11) to be written compactly as

$$\dot{\boldsymbol{\zeta}}_k = \hat{\boldsymbol{A}}_k \boldsymbol{\zeta}_k + \hat{\boldsymbol{B}}_k \mathbf{u}_k. \tag{4.12}$$

Furthermore, if the augmented system is controllable, i.e., the controllability matrix

$$\hat{\boldsymbol{U}}_C = \begin{bmatrix} \hat{\boldsymbol{B}}_k & \hat{\boldsymbol{A}}_k \hat{\boldsymbol{B}}_k & \hat{\boldsymbol{A}}_k^2 \hat{\boldsymbol{B}}_k & \dots & \hat{\boldsymbol{A}}_k^{(n-1)} \hat{\boldsymbol{B}}_k \end{bmatrix} \tag{4.13}$$

is full rank, then, using the feedback control law

$$\mathbf{u}_k = -\hat{\boldsymbol{K}}_k \boldsymbol{\zeta}_k,$$

there exists a $\hat{\boldsymbol{K}}_k$ such that the eigenvalues of the closed-loop system

$$\dot{\boldsymbol{\zeta}}_k = \big(\hat{\boldsymbol{A}}_k - \hat{\boldsymbol{B}}_k \hat{\boldsymbol{K}}_k\big)\boldsymbol{\zeta}_k \tag{4.14}$$

has the property

$$\Re\{\lambda_i(\hat{\boldsymbol{A}}_k - \hat{\boldsymbol{B}}_k \hat{\boldsymbol{K}}_k)\} = \hat{\lambda}_{des,i} < 0 \quad \forall i = 1, \dots, 11.$$

where $\hat{\lambda}_{des,i}$ is the i^{th} component of $\hat{\boldsymbol{\lambda}}_{des}$. Thus, the linearized closed-loop augmented system is stable and asymptotically converges to the point $\boldsymbol{\zeta}_k = \mathbf{0}$. In other words, the closed-loop nonlinear system is stable and asymptotically converges to the trim state and and desired flight condition. The controllability of the matrix pair $[\hat{\boldsymbol{A}}_k,\ \hat{\boldsymbol{B}}_k]$ is dependent on the size of $\mathbf{u}_k$ and the controllability of $[\boldsymbol{A}_k,\ \boldsymbol{B}_k]$: the size of $\mathbf{u}_k$ must be greater than the size of $\boldsymbol{y}_k$ and, given the definition of $\boldsymbol{C}_k$ above, results show that if $[\boldsymbol{A}_k,\ \boldsymbol{B}_k]$ are controllable, then $[\hat{\boldsymbol{A}}_k,\ \hat{\boldsymbol{B}}_k]$ are controllable.

4.3.2 Nonlinear PID Control

Extending the linear analysis to the time-varying nonlinear aircraft yields the proportional-integral-derivative (PID) nonlinear controller

$$\begin{aligned}\boldsymbol{\mu}(h(t),\, v^*(t),\, \dot{h}^*(t),\, \dot{\psi}^*(t)) &= \boldsymbol{\mu}^*(t) - \hat{\mathbf{K}}(t)\boldsymbol{\zeta}(t) \\ &= \boldsymbol{\mu}^*(t) - \mathbf{K}_x(t)\mathbf{x}(t) + \mathbf{K}_\xi(t)\int_{t_0}^{t}\mathbf{y}(\tau)d\tau \\ &= \boldsymbol{\mu}^*(t) - \mathbf{K}_x(t)\big(\bar{\mathbf{z}}(t) - \bar{\mathbf{z}}^*(t)\big) \\ &\quad + \mathbf{K}_\xi(t)\int_{t_0}^{t}\begin{bmatrix} v(\tau) - v^*(\tau) \\ \dot{h}(\tau) - \dot{h}^*(\tau) \\ \dot{\psi}(\tau) - \dot{\psi}^*(\tau)\end{bmatrix} d\tau\end{aligned} \tag{4.15}$$

where $\hat{\boldsymbol{K}}_k = [\boldsymbol{K}_x(t) \quad \boldsymbol{K}_\xi(t)]$ and t_0 is transition start time. (4.15) works analogously to (4.6): at each instant in time, the supplied flight condition defines the feed-forward and feedback controller parameters, with the only exception the introduction of an additional integral term. Again, (4.15) can be represented by the block diagram shown in Figure 4.11 where $\mathbf{h}(\mathbf{z})$ represents the nonlinear function measuring the actual flight condition:

$$\mathbf{h}(\mathbf{z}) = \begin{bmatrix} v \\ \dot{h} = \cos\alpha\cos\beta\sin\theta - \sin\phi\sin\beta\cos\theta - \cos\phi\sin\alpha\cos\beta\cos\theta \\ \dot{\psi} = \dfrac{q\sin\phi + r\cos\phi}{\cos\theta}\end{bmatrix} \tag{4.16}$$

The improvement provided by implementing (4.15) is shown in Figure 4.12, which shows the output response and control signal for the example transition of the previous sections. The response in Figure 4.12 was produced by scheduling C_{PID}, in the same way as C_{PD}, now with the $\boldsymbol{K}_\xi$ database. Whereas the PD controlled relied on interpolated estimates of the trim state, the desired flight condition is included resulting in much smoother transients than in the PD case.

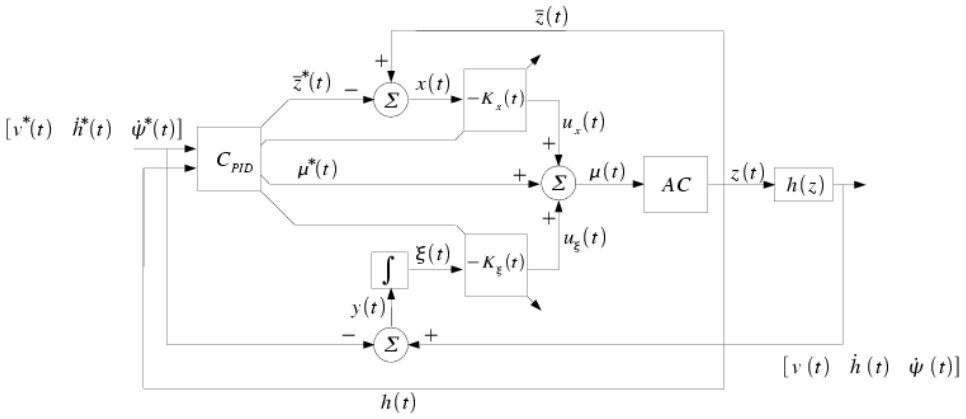


Figure 4.11: PID Control Block Diagram

The flight path following capabilities of C_{PID} are also improved, as shown in Figure 4.13. The steady-state error between the desired and actual altitude and heading angle have been eliminated; a constant steady-state error remains in the x and y position variables.

One thing that should be noted is that the linear analysis used to design the controller above does not equate similar properties to the full nonlinear system. Most importantly, the stability of (4.14) does not guarantee the stability of the closed-loop system occurring from implementation of (4.15). The controller design assumes time-invariant linear dynamics, when in fact, the linearized aircraft dynamics vary over the changing desired flight condition, and hence, over time. Rather than tackling the full time-varying analysis to prove closed-loop stability, this work assumes the empirical stability properties are sufficient and instead focuses on the good performance and tracking characteristics seen from implementing the PID controller.

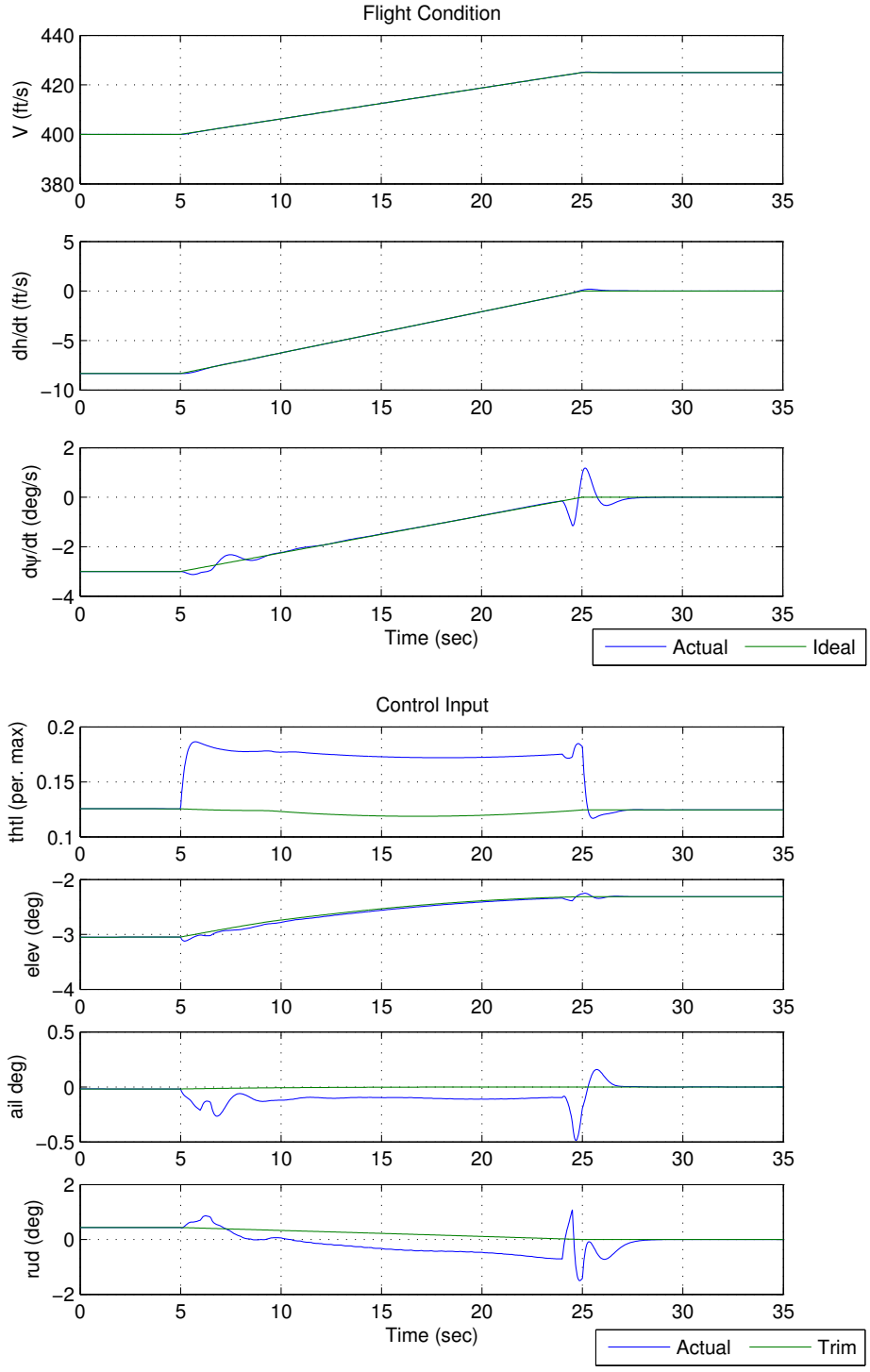


Figure 4.12: PID Controller Response

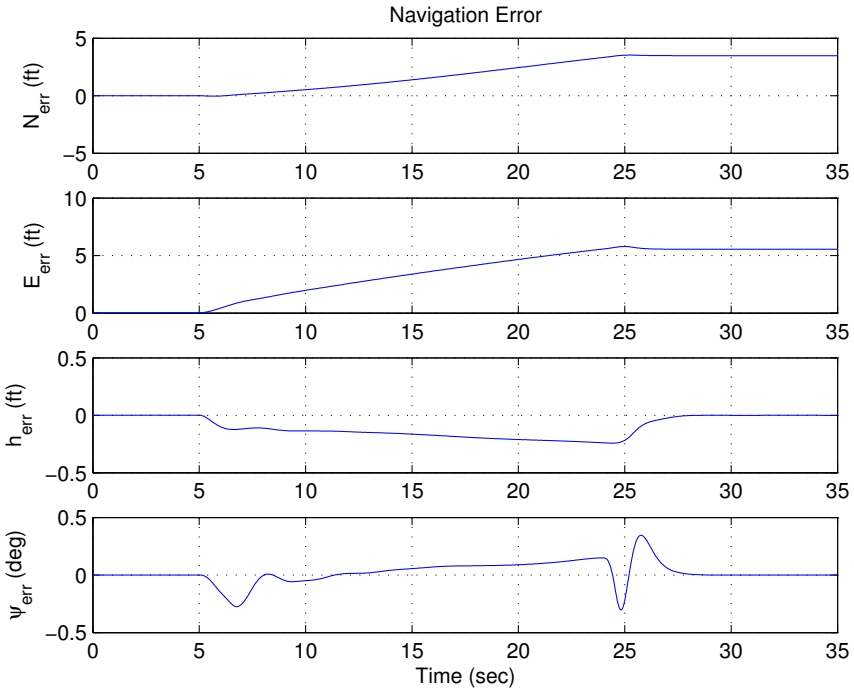


Figure 4.13: Ground Track Error using PID Controller

4.4 Transition Flight Path Displacement

The transition maps $\boldsymbol{G}_{i-1,i}$ do not have exact analytical or accurate approximate solutions. To a first approximation, $\boldsymbol{G}_{i-1,i}$ can be approximated by the ideal flight path displacement above, though the steady-state error present in x and y would produce equivalent errors between the simplified aircraft kinematic model and the actual aircraft response. Such errors will also be compounded over numerous segments, producing even larger deviations.

Alternatively, these transition maps can be quantified using a time propagation (numerical simulation) of the full nonlinear model. Suppose the transition (4.1) is numerically simulated between times t_1 and t_2, where t_1 marks when the transition is initiated and by t_2, all the transients have died away, producing the time histories $\mathbf{p}(t)$ and $\psi(t)$. The transition map $\boldsymbol{G}_{i,j}$ can then be quantified via

$$\boldsymbol{G}_{i,j} = \begin{bmatrix} \boldsymbol{R}_{\Delta\psi}^T & \boldsymbol{R}_{\psi_1}^T(\mathbf{p}(t_2) - \mathbf{p}(t_1)) \\ 0 & 1 \end{bmatrix}, \tag{4.17}$$

where

$$\boldsymbol{R}_{\Delta\psi} = \begin{bmatrix} \cos(\psi(t_2) - \psi(t_1)) & \sin(\psi(t_2) - \psi(t_1)) & 0 \\ -\sin(\psi(t_2) - \psi(t_1)) & \cos(\psi(t_2) - \psi(t_1)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.18}$$

$$\boldsymbol{R}_{\psi_1} = \begin{bmatrix} \cos(\psi(t_1)) & \sin(\psi(t_1)) & 0 \\ -\sin(\psi(t_1)) & \cos(\psi(t_1)) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.19}$$

Performing similar numerical simulations over the spectrum of different trimmed flight conditions yields the maneuver database M of Section 3.5. While the choice of t_2 is arbitrary, it in general occurs sometime after the commanded transition.

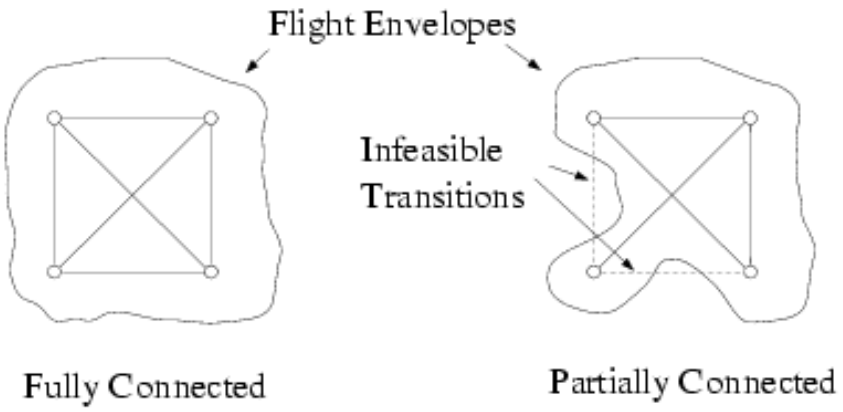


Figure 4.14: Example Transition Maps

This additional coast time is required to allow the integral controller to converge on the now steady flight condition and can be defined as $t_c = t_2 - (t_1 - \Delta t)$. For consistency, every transition in M is performed with identical Δt and t_c.

4.4.1 Maneuver Database Connectedness

It has thus far been assumed that each transition in M is feasible. If so, the maneuver database is fully connected. However, this, is not always the case. In fact, most M are non-convex, i.e., partially-connected, potentially consisting of infeasible transitions. In general, there exist two instances where transitions are infeasible: (1) the desired transition passes through an infeasible region of the flight envelope or (2) the transition results in actuator saturation.

The first case is a result of the type of interpolation used in the controller. At each instant in time, the PID controller uses trim state $\bar{\mathbf{z}}^*(t)$ generated by C_{PID}. Since the supplied flight condition to C_{PID} varies according to (4.5), transitions

can also be defined as straight lines connecting feasible trim states in the flight envelope space. If this straight line passes through an infeasible region of the flight envelope, then at some point along the transition, the desired trim state $\bar{\mathbf{z}}^*(t)$ returned by C_{PID} will cease to be defined. As a result, the controller will have no reference state, a condition that leads ultimately to instability. A schematic, example two-dimensional case of a transition traversing an infeasible part of the flight envelope is provided in Figure 4.14.

The second source of infeasible transitions, actuator saturation, is a bit more subtle and varies significantly from failure to failure. Imposing a limit on Δt and t_c for a given M implicitly defines a limit on the maximum *length*—the distance between two trim states in the trim database—a feasible transition can have. Performing longer transitions over the same Δt requires more control input than if Δt were increased. As a result, some large transitions cannot be performed without saturating the actuators. Unlike the transitions outside the flight envelope, detecting infeasible transitions due to control saturation is impossible analytically. Moreover, control saturation is directionally-dependent: A transition from i to j may satisfy the control limitations while a transition from j to i is infeasible. A transition may also be altitude-dependent, feasible at h_i, but not at h_j.

In both cases, during the compilation of M, two checks are performed on each simulated transition: (1) that the desired terminal trim state has been reached and (2) that the actuator settings remain within their physical limits. Those transitions that violate either of these rules are denoted in M by an instantaneous transition, or, more explicitly, if a transition is infeasible at an altitude h^*, then

$$\boldsymbol{G}_{i,j} = M(h^*,\, v_i^*,\, \dot{h}_i^*,\, \dot{\psi}_i^*,\, v_j^*,\, \dot{h}_j^*,\, \dot{\psi}_j^*) = \boldsymbol{I}_4 \tag{4.20}$$

where $\boldsymbol{I}_4$ is the 4×4 identity matrix. (4.20) works as a flag indicating that a particular sequence is infeasible. This flag will later alert the planning algorithm that a candidate plan is not feasible and should not be explored.

Chapter 5

Trajectory Planning

This chapter describes a flight planning algorithm capable of identifying feasible paths in the inertial space connecting an aircraft at some initial post-failure state with a desired landing site. Given a set of feasible trimmed flight conditions and transitions between them, the planning algorithm must be able to effectively sequence segments together producing a trajectory having the desired initial and terminal position and heading values. However, identification of this sequence is a nontrivial task given the size of typical trim and transition databases. To enable real-time trajectory planning, the algorithm finds feasible rather than optimal flight plans allowing the number of states considered by the planner to be significantly reduced. Once reduced trim and transition databases have been specified, the resulting trim and maneuver sequences can be exhaustively searched, using the kinematic aircraft model to validate each sequence.

This chapter contains two sections. Section 5.1 describes the method by which the full trim database is reduced to a tractable size, and Section 5.2 describes the hybrid path planning algorithm designed to efficiently sequence trimmed flight segments into a feasible post-failure flight plan.

5.1 Database Reduction

The full set of controllable state in the trim database can be represented as the set

$$D = \{(h_k^*,\ v_k^*,\ \dot{h}_k^*,\ \dot{\psi}_k^*)\}_{k=1,\ldots,N_D} \tag{5.1}$$

where N_D is the total number of points contained within D. Since altitude cannot be independently specified, D can be contracted over altitude to produce the flight condition database D', the intersection of all three-dimensional altitude slices, from h_0 to h_n, in the trim database:

$$D' = \bigcap_{i=0}^{n} D(h_i^*,\ v_k^*,\ \dot{h}_k^*,\ \dot{\psi}_k^*) = \{(v_k^*,\ \dot{h}_k^*,\ \dot{\psi}_k^*)\}_{k=1,\ldots,N_{D'}} \tag{5.2}$$

where $N_{D'} < N_D$. Further contraction of the database can be accomplished by removing additional points, retaining a sufficient set approximately spanning the flight envelope, enabling the planner to find solutions if they exist. The new, contracted database $\tilde{D}$ can be represented as

$$\tilde{D} = \{(v_k^*,\ \dot{h}_k^*,\ \dot{\psi}_k^*)\}_{k=1,\ldots,N_{\tilde{D}}} \tag{5.3}$$

where, by definition, $\tilde{D} \subset D'$ and $N_{\tilde{D}}$ is the size of the contracted database such that $N_{\tilde{D}} \ll N_D$.

5.1.1 Heuristic Reduction

While the reduction from $D \rightarrow D'$ is rigorously defined, a heuristic reduction process is used to reduce $D' \rightarrow \tilde{D}$. The points selected should represent the suite of possible flight condition classes, that is, flight conditions where $-/0/+$ turn and climb/descent combinations are possible.

One possible heuristic reduction process would be to define a series of nested, concentric cube inside the three-dimensional volume defined by D' as shown in

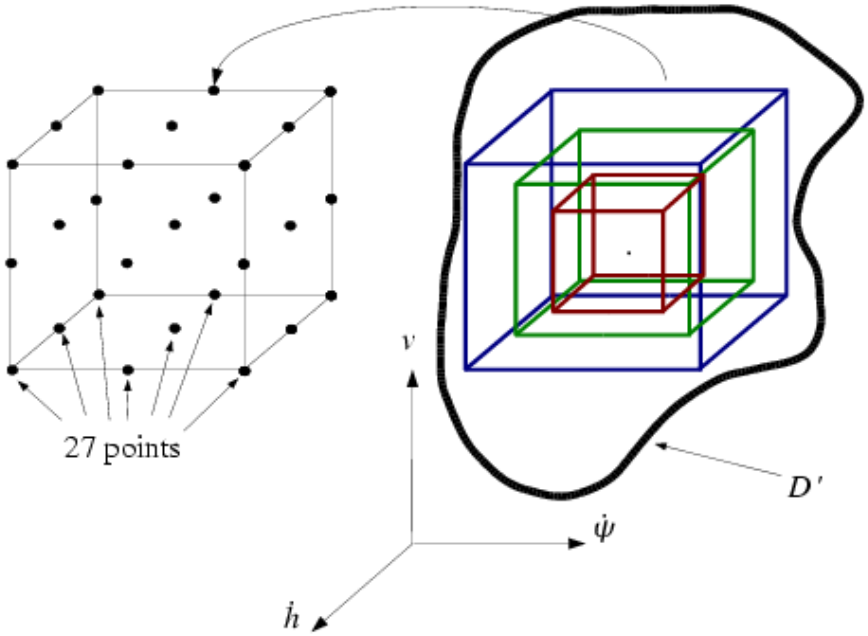


Figure 5.1: Nested Cubes Method for Database Reduction

Figure 5.1. Each cube is aligned so that each axis of the cube is aligned with one of the dimensions of D' and defines 27 different points in the airspeed-climb rate-turn rate space—26 points on the surface of the cube, and the cube's center. Each three-dimensional point corresponds to a particular flight condition that is included in the reduced $\tilde{D}$. By allowing multiple cubes, each of a different size, multiple *layers* of the feasible flight envelope can be included in $\tilde{D}$. The largest cube represents the set of extreme flight conditions considered by the path planner; the lengths of the cube's sides are chosen so that the entire volume of the cube is just inside the boundary of D'. Again, the trade-off between database size and search speed requires a choice of 2 or 3 nested cubes. Design expertise of the specific failure and aircraft will guide the choices for the number of cubes, as well as the size for each.

An alternate, yet similar, technique would consist of defining squares in the D' space, instead of cubes. With this approach, slices of the D' volume are taken over a discrete set of airspeeds. Within each slice, a square is placed that encompasses the range of climb and turn rates of the two-dimensional slice. Over the discrete range of airspeeds, the combined three-dimensional shape better matches D'; using the method described above, the irregularness of D' is not accounted for leading to an unnecessarily conservative choice for $\tilde{D}$. A graphical representation of the this method is shown in Figure 5.2. Each square would define 9 points in the D' space—8 along the edge and corners and one at the center. As each square is defined, and stacked on each other, the outer 8 points can, in a sense, be seen as connected, defining with each slice a larger, multi-sided, flat-faced three-dimensional shape outlining $\tilde{D}$. Additional points can be included in $\tilde{D}$ by incorporating both techniques mentioned here; nesting squares at each selected airspeed includes more of the internal shape of D' while providing adequate coverage of the velocity range.

Because this method better captures the shape of D', the boundary of the combined volume of squares is more likely to intersect the boundary of D'. The lines connecting these varied points correspond to the transitions between trim states, and as mentioned before, the intersection of these lines with the boundary of D' represent transitions that occur outside the feasible flight regime. Thus, to ensure the minimal number of infeasible transitions, the selection of trim values away from the flight envelope boundary is preferred.

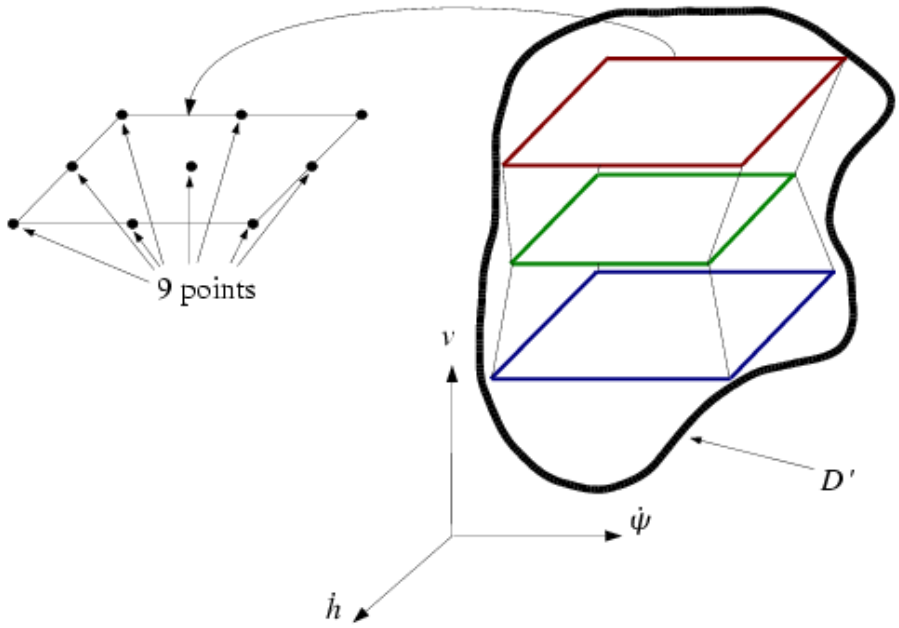


Figure 5.2: Stacked Squares Method for Database Reduction

5.1.2 Database Reduction: An Example

As an example, consider rudder failure sea-level trim database shown in Figure 5.3. Notice that the aircraft can fly straight and make both left and right turns at various negative and positive climb rates. Thus, appropriate choices for the trim states in $\tilde{D}$ could include all possible combinations of positive, negative, and zero climb rates and positive, negative, and zero turn rates, over a range of reasonable airspeeds. Furthermore, if Figure 5.3 represents D', example values for reasonable $\tilde{D}$ states are shown in Table 5.1. Each row of the table represents an airspeed slice. For each airspeed, the trim states in $\tilde{D}$ consists of all possible combinations the climb rates (second column) and turn rates (third column) shown.

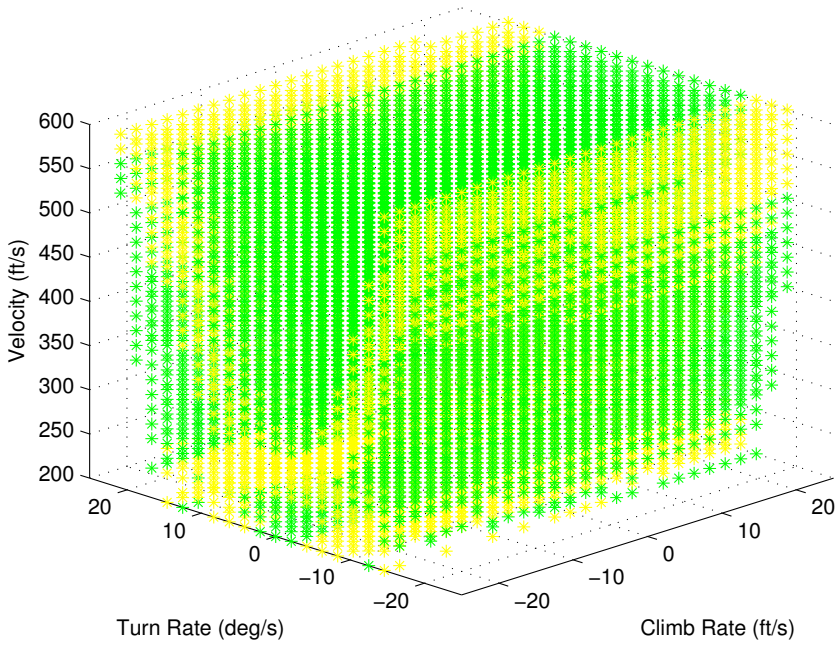


Figure 5.3: F-16 0° Rudder Failure Trim Database at Sea-Level

5.1.3 Reduced Kinematic Databases

Once a suitable reduced flight condition database has been selected, the companion reduced pseudo-body velocity database $\tilde{T}$ and reduced transition database $\tilde{M}$ can be compiled from the flight conditions in $\tilde{D}$ and a set of altitudes in the range $[h_0, h_n]$. If this set was a subset of the original altitude discretization that created the trim database, then $\tilde{T} \in T$ and $\tilde{M} \in M$, though, this need not be the case. The range of altitude producing $\tilde{T}$ and $\tilde{M}$ can be finer than in the original trim database to increase the accuracy of the interpolation routine that approximates intermediate pseudo-body velocities and transition mappings.

One issue that could arise in compiling $\tilde{M}$ is the reduction in the ratio of feasible to infeasible transitions. That is, the definition of $\tilde{D}$ could eliminate many

Table 5.1: Example Values for $\tilde{D}$ for Nominal F-16 Aircraft

Airspeed (ft/s)	Climb Rate (ft/s)	Turn Rate (deg/s)
250	0, ±5, ±10	0, ±1, ±3
300	0, ±7.5, ±15	0, ±1.5, ±4.5
400	0, ±10, ±20	0, ±2, ±6
600	0, ±15, ±30	0, ±3, ±9

of the feasible transitions initially found in M. Such a skew in the transition mappings would hinder the planner in finding valid flight plans; the selection of $\tilde{D}$ in this case defines a severely limited search space. If the transition database is deemed too sparse, the definition of $\tilde{D}$ can be altered until the resulting $\tilde{M}$ is sufficiently full. Increasing the number of points in $\tilde{D}$ would almost guarantee a corresponding increase in the feasible transition ratio, although it would negatively impact the solution calculation speed. Similarly, the construction of $\tilde{D}$ could be scrutinized until the a suitable $\tilde{M}$ results.

5.1.4 Feasible Path Existence

Whereas the discussion thus far has focused on reducing the trim database to a tractable size to increase planner efficiency, decreasing the number of segments of examined landing plans also improves planner speed. Additionally, minimizing N facilitates management by pilots and air traffic control. Because fewer segments also decrease the number of possible solutions, it is desirable to find the minimum number of segments guaranteeing feasible trajectories while maximizing planning efficiency.

Typically, for a nominal aircraft, the minimum number of flight segments required to produce a feasible landing plan is four: two straight and two turning segments. Therefore, the minimal path for a nominal aircraft would include a turn and fly (straight) segment toward the final intercept path, then a turn onto final (straight) [7]. 3-segment minimal paths can also be constructed for an aircraft with no constraint on climb rate. However, the constraint to stay within the feasible post-failure flight envelope, requires an additional segment to compensate. Thus, with full positive and negative capabilities for climb and turn rate, a minimum of four segments are required.

As more restricted example, consider a case in which only turns in one direction are possible. The minimum N required then becomes a function of the maximum and minimum feasible turn rates. In particular, a minimum flight plan consists of flight segments alternating between the maximum and minimum allowable turn rates. However, if positive and negative climb rates are also feasible, then there exists a finite number of segments capable of transitioning the aircraft between any two position and heading locations [29]. Such a complicated flight path is nonintuitive and would try most pilots concentration necessitating some form of avionic assistance.

In the opposite case, where both left and right turns, as well as straight flight, are possible but only negative flight is feasible (only positive flight not a possible failure scenario), then a minimal flight plan still consists of only four segments, though, feasible paths only for certain terminal position and heading combinations. This situation must be addressed when selecting desired landing sites, and would handled by a coupled site selector. Such an analysis is beyond the scope of this thesis, though previous work [7] has defined a landing site search

(LSS) algorithm applied to the case of a total loss of thrust. Such an algorithm need only be minimally augmented to handle more general failure modes.

A final concern when selecting flight plan lengths is the location of the initial flight condition of the aircraft at $\boldsymbol{F}_0$. In some cases, where straight flight is feasible, the structure of $\tilde{M}$ requires multiple sequences to reach this flight condition. As a result, the planner will more than likely require additional segments to compensate for the potentially necessary maneuvering through the feasible flight condition space. As with all the scenarios addressed in this section, selecting the number of segments the planner will use in this case could by automated, though in this work, N is set *a priori* for all examples.

5.2 Trajectory Planning

The purpose of the planner is to identify a sequence of trimmed flight conditions from $\tilde{D}$ that allows the aircraft to reach its destination—the desired landing site—with the correct heading. As described previously, this is equivalent to solving the inverse aircraft kinematic problem for which no general solution exists because of the mixture of continuous and discrete quantities. Using a purely discrete method, such dynamic programming [36–38] where Δt_i are discretized, would increase the computational complexity of the problem. Instead, the approach taken in this thesis is to perform a mixed continuous/discrete optimization: a discrete search over possible flight condition sequences where at each iteration a continuous optimization is performed to select appropriate values for Δt_i.

5.2.1 Planning Notation

To facilitate the discussion of the planning algorithm, each flight segment can be specified as

$$s_i = (v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*,\ \Delta t_i) \tag{5.4}$$

where the triple $(v_i^*, \dot{h}_i^*, \dot{\psi}_i^*) \in \tilde{D}$ and Δt_i is the time trim segment i is maintained. A plan then can be expressed as a sequence of N trim segments s_i:

$$P = \{s_i\}_{i=1}^N = \{(v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*,\ \Delta t_i)\}_{i=1}^N. \tag{5.5}$$

A *candidate* plan $\tilde{P}$ is a partially instantiated plan consisting of valid values for the flight condition for each segment, but without a specific duration specified. More specifically, a candidate plan $\tilde{P}$ can be expressed by

$$\tilde{P} = \{(v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*)\}_{i=1}^N. \tag{5.6}$$

5.2.2 Planning Algorithm

The complete planning algorithm is shown in Figure 5.4. *Planner* takes as variables the initial aircraft location and heading, $(\mathbf{p}_0,\ \psi_0)$, the desired landing site and heading, $(\mathbf{p}_{des},\ \psi_{des})$, the number of segments N, the type of failure $Fail$, and the allotted planning time t_{plan}. Upon initialization, the algorithm calls $ReadInData$ which accesses failure-specific databases $\tilde{D}$, $\tilde{T}$, $\tilde{M}$, and stores them in memory. The number of flight conditions in $\tilde{D}$ is determined and used to compute the number i_{max} of candidate paths contained in the combinatorial set defined by $\tilde{D}$ and N. The optimal plan list L is also initialized.

The algorithm examines all possible sequences of N flight conditions in $\tilde{D}$, within the alloted planning time t_{plan}, to identify a minimum-cost solution connecting $(\mathbf{p}_0,\ \psi_0)$ to the landing site. The procedure $GetCanidatePlan$ uses the

Algorithm: PLANNER($\mathbf{p}_0, \psi_0, \mathbf{p}_{des}, \psi_{des}, N, Fail, t_{plan}$)

$(\tilde{D}, \tilde{T}, \tilde{M}) \leftarrow$ READINDATA($Fail$)
$N_{\tilde{D}} \leftarrow$ SIZEOF($\tilde{D}$)
$i_{max} \leftarrow N_{\tilde{D}}^{N}, i \leftarrow 0$
$L \leftarrow \emptyset$
while $(i < i_{max})$ **and** $(\delta t < t_{plan})$
do
- $\tilde{P}_i \leftarrow$ GETCANDIDATEPLAN($i, \tilde{D}, N$)
- **if false** $\leftarrow$ VALIDATESEQUENCE($\tilde{P}_i, \tilde{M}$)
  - **then** $i \leftarrow i + 1$; **continue**
- **if false** $\leftarrow$ PLANCONSTRAINTS($\tilde{P}_i$)
  - **then** $i \leftarrow i + 1$; **continue**
- $(J^*_{plan}, P_i) \leftarrow$ GETDURATIONS($\tilde{P}_i, \tilde{T}, \tilde{M}, \mathbf{p}_0, \psi_0, \mathbf{p}_{des}, \psi_{des}$)
- **if** $J^*_{plan} \leq \epsilon_{plan}$
  - **then**
    - **if false** $\leftarrow$ CHECKALTITUDE(P_i)
      - **then** $i \leftarrow i + 1$; **continue**
    - $l \leftarrow$ COMPUTELENGTH(P_i)
    - SORTLIST(L, P_i)
    - $i \leftarrow i + 1$; **continue**
  - **else** $i \leftarrow i + 1$; **continue**

return (L)

Figure 5.4: *Planner* Algorithm

current iteration count as a marker to produce the i^{th} candidate plan $\tilde{P}_i$. In $ValidateSequence$, each $\tilde{P}_i$ is tested to make sure all necessary transitions are feasible with respect to (4.20), at every altitude. The procedure $PlanConstraints$ then tests $\tilde{P}_i$ against additional user-defined constraints. A typical constraint is that the aircraft must be descending or level during its final approach segment. Additional constraints will typically further restrict the terminal segment by either constraining $\dot{h}_N$ and/or $\dot{\psi}_N$.

Once a candidate plan has been validated, $GetDurations$ computes the $\{\Delta t_i\}_{i=1}^N$ using a numerical optimization algorithm to minimize the cost function

$$J_{plan}(\{\boldsymbol{G}_{i-1,i},\ \boldsymbol{G}_i\}_{i=1}^N) = \|\boldsymbol{F}_0 \prod_{i=1}^{N} \boldsymbol{G}_{i-1,i}\boldsymbol{G}(v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*, \Delta t_i) - \boldsymbol{F}_{des}\|_2 \qquad (5.7)$$

over the N continuous variables $\{\Delta t_i\}_{i=1}^N$, where the shorthand $\boldsymbol{G}_i$ was used for $\boldsymbol{G}(v_i^*,\ \dot{h}_i^*,\ \dot{\psi}_i^*,\ \Delta t_i)$, $\boldsymbol{F}_{des}$ is defined using $(\mathbf{p}_{des},\ \psi_{des})$, and the weight w is used to balance position and heading penalties. At each iteration, (5.7) is computed from the kinematic aircraft model to predict the final position and heading $(\mathbf{p}_N,\ \psi_N)$ for flight condition sequence $\tilde{P}$ and the current iteration of $\{\Delta t_i\}_{i=1}^N$. Once the numerical optimizer has effectively minimized (5.7), $GetDurations$ returns J_{plan}^* where

$$J_{plan}^* = \min_{\{\boldsymbol{G}_{i-1,i},\, \boldsymbol{G}_i\}_{i=1}^N} \{J_{plan}\} \qquad (5.8)$$

and stores the fully instantiated plan in P_i. Whereas most optimization routines available in standard numerical libraries [39] can suitably minimize J_{plan}, previous results [40] have shown that the Nelder-Mead simplex algorithm provides accurate results given a relatively complex cost function (5.7), thus the Nelder-Mead was adopted for this work, with initial values for $\{\Delta t_i\}_{i=1}^N$ set to zero.

A path is considered considered acceptable if

$$J^*_{plan} \leq \epsilon_{plan} \tag{5.9}$$

where ϵ_{plan} is a small, positive scalar representing the threshold between acceptable and non-acceptable solutions. When an acceptable solution is identified, P_i is checked by *CheckAltitude* for altitude constraint satisfaction, that is,

$$h_i > h_{min} \quad \forall i = 1, \ldots, N. \tag{5.10}$$

If this single constraint is satisfied, the plan is then added to the plan list L. Otherwise, the solution is abandoned and the algorithm builds a new candidate plan from the feasible trim state list. To identify the optimal flight plan, P_i in L are graded by *SortList* according to their total duration

$$\Delta T = \sum_{i=1}^{N} \Delta t_i, \tag{5.11}$$

and sorted by increasing ΔT. Once all trim sequence combinations have been examined, the plan list L is returned to the flight management system and/or pilot for implementation.

The weight w and threshold ϵ_{plan} are independent design parameters allowing the adjustment of the acceptable solution criterion. By adjusting w, the relative penalty between one foot of position error and one radian of heading error can be refined. Since large ψ errors could result in the aircraft running off the side of the runway, errors in heading should be weighted much more than errors in position. Thus, the case study presented in this work uses $w = 1000$ so that one foot of position error is equivalent to 0.001 radians (or about 0.06 deg) of heading error. Whereas w equates position and heading error, ϵ_{plan} specifies the maximum weighted error magnitude allowed in acceptable solutions. In the case

study to follow, the value $\epsilon_{plan} = 1$ was used. This means that a solution with one foot of position error magnitude and perfect heading is acceptable. Similarly, a solution with a perfect position and 0.001 radians of heading error would also be acceptable. As a result, w also describes how much decrease in the heading error is required accept an equal increase in position error.

5.2.3 Any-time Emergency Flight Planning

As a final note, to run to completion, the planner must exhaustively search through $N_{\tilde{D}}^{N}$ iterations, each of which requires an embedded numerical optimization. Although this path planner can identify a shortest-duration solution, in practice, it cannot efficiently find this solution within the allotted time limit t_{plan}. However, since feasible solutions are returned to L as quickly as they are found, L can be returned to the parent function after the time t_{plan}.

Such an *anytime* [41–43] adaptive flight planner, which trades solution quality for required computation time, is shown in Figure 5.5 . Upon the determination of the failure type, the aircraft is commanded to a constant holding pattern ($\dot{h} = 0$) or spiral descent from high altitude. Once at this flight condition, the aircraft's position is projected to some known waypoint ($\mathbf{p}_0$, ψ_0), designated as the initial location in the flight planner. Using the projected waypoint ($\mathbf{p}_0$, ψ_0), the coupled LSS algorithm can select a feasible landing site from an integrated airport database. The number of segments used in the flight plan is determined as well as the allotted planning time t_{plan}. From here, the flight planner is called and returns the current acceptable list L after time t_{plan}. The current most optimal plan P_{min} is selected from the list and executed.

However, if the list L is empty—when the planner cannot find a solution

1. Determine type of failure $Fail$.
2. Command aircraft to safe holding/descent pattern (v_0^*, $\dot{h}_0^*$, $\dot{\psi}_0^*$).
3. Compute current state ($\mathbf{p}_{-1}$, ψ_{-1}).
4. Use kinematic aircraft model to project aircraft to initial location ($\mathbf{p}_0$, ψ_0).
5. Find feasible landing site via LSS ($\mathbf{p}_{des}$, ψ_{des}).
6. Set N based on failure type.
7. Determine allowed planning time t_{plan}.
8. $L \leftarrow$ PLANNER($\mathbf{p}_0$, ψ_0, $\mathbf{p}_{des}$, ψ_{des}, N, $Fail$, t_{plan})
   while $L = \emptyset$
   do $\begin{cases} (\mathbf{p}_0, \psi_0, t_{plan}) \leftarrow \text{LOITER}() \\ L \leftarrow \text{PLANNER}(\mathbf{p}_0, \psi_0, \mathbf{p}_{des}, \psi_{des}, N, Fail, t_{plan}) \end{cases}$
   $P_{min} \leftarrow L(1)$
   execute P_{min}

Figure 5.5: Outline for Adaptive Flight Planner

within the time t_{plan}—then the *Loiter* procedure is called and the aircraft is commanded to a holding pattern, and the initial waypoint ($\mathbf{p}_0$, ψ_0) and planning time t_{plan} are recomputed and passed to *Planner* again. Instead of implementing the full adaptive flight planner as outlined in Figure 5.5, this thesis examines solely flight planner functions and assumes that t_{plan} is sufficiently long to allow the exhaustive search of all candidate plans.

Chapter 6

F-16 Case Study

This chapter presents a series of failure cases for the nonlinear F-16 aircraft model from Section 2.3. For each scenario, full and reduced trim databases and transition map are shown, as well as example post-failure trajectories generated by the planner. The examples provided in this chapter highlight the non-intuitive aircraft orientations required to maintain flight, as well as the complicated movement through the post-failure flight envelope necessary to plan feasible trajectories to specified landing sites. All presented results were generated on a 2.00GHz Intel® Pentium® 4.

The failure scenarios examined in this thesis reprsent failure of one of the F-16's lateral control actuators: the aileron and rudder. It will be assumed that the control surface jam failures occur, situations that may result from airframe damage, mechanical binding, or even the loss of hydraulic pressure. In this type of failure, the uncontrollable actuator is stuck at a fixed deflection, which, when stuck at a non-zero angle, contributes significant non-zero forces and torques.

6.1 F-16 Controller Implementation

An F-16 controller was implemented to develop the transition databases and to verify feasibility of planned post-failure trajectories. Due to the high-computational cost of interpolating during each controller cycle, the PID controller developed in Section 4.3.2 was simplified to require only one-dimensional interpolations instead of four-dimensional. This was done by interpolating over the trim states instead of over the flight condition, as originally stated. This method is analogous to the trim state open-loop control law discussed in Section 4.1.3.

While this modified controller structure allows for faster cycles times, it suffers from potential mismatches between the desired flight condition used in output feedback controller (the integral controller) and the trim states used as state feedback. Transitions between trim states were previously defined to be a linear change in the flight condition between the those of the initial and terminal trim states. The resulting change of the trim states over these transition, however, can vary nonlinearly. As a result, in the above interpolated trim state controller, when the output feedback controller is attempting to regulate the desired flight condition, it is acting against the state feedback controller attempting to regulate the interpolated trim state. The result, if the mismatch is sufficiently large enough, is loss of stability. Such mismatches can be avoided if the jumps in the transition maps are small; approximating how small small needs to be is done on a case by case basis.

Table 6.1: Discrete Trim Database Flight Conditions

	min	max	Δ	No. Pts.
h (ft)	0	30,000	10,000	4
v (ft/s)	200	600	16.667	25
$\dot{h}$ (ft/s)	-25	25	2.083	25
$\dot{\psi}$ (deg/s)	-25	25	2.083	25
Total Data Points				62,500

6.2 Rudder Failure

6.2.1 Trim Database Calculation

Using the steps outlined in Chapter 3, aircraft performance was characterized for 0°, 15°, and 30° jammed rudder failures over the four-dimensional grid formed from combinations of values from Table 6.1. To check for feasibility, a value of $\epsilon_{trim} = 10^{-7}$ was used for (3.10).

At each feasible trim state, the linearized dynamics were calculated using (3.14) with a value $\epsilon = .01$ for airspeed and $\epsilon_i = .0001$ used for all other parameters. This linear system was then analyzed to characterize stability and controllability near each trim point using the method from Section 3.2. While stability is defined by means of (3.15), a more useful criterion is

$$\Re\{\lambda_i(\boldsymbol{A}_k)\} < -\epsilon_{stab} \quad \forall i = 1, \ldots, 8$$

for some small $\epsilon_{stab} > 0$. This numerical buffer around zero helps protect the stability calculation against inaccuracies resulting from numerical round-off error. Similarly, when numerically computing (3.18), a numerical buffer for the rank

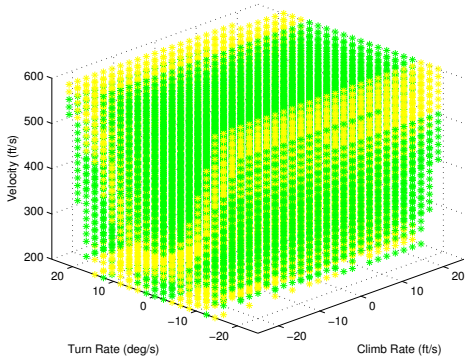
calculation is also needed. One method commonly used to compute the matrix is the singular value decomposition. The rank of U_C is the number of non-zero singular values of U_C [44]. Therefore, a more practical condition for controllability is that

$$\sigma_i(U_C) < -\epsilon_{ctrb} \quad \forall i = 1, \ldots, 8$$

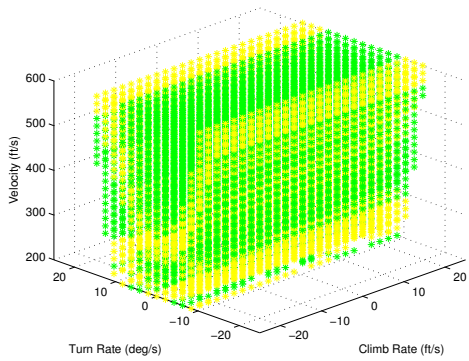
where $\sigma_i(A)$ is the i^{th} singular value of A and ϵ_{ctrb} is a small positive scalar. For the trim databases presented in this chapter, values of $\epsilon_{stab} = 10^{-3}$ and $\epsilon_{ctrb} = 10^{-12}$ were used and correspond to the built-in defaults for the numerical algorithms used.

The results of the above calculations for each failure trim database are shown in the altitude slices of Figures 6.1-6.3. Each entry in the database has been labelled according to its stability and controllability properties: green states are both stable and controllable, yellow states are unstable but controllable, and red states are both unstable and uncontrollable. Additionally, those states that are infeasible at each failure were omitted in the plots to improve clarity.

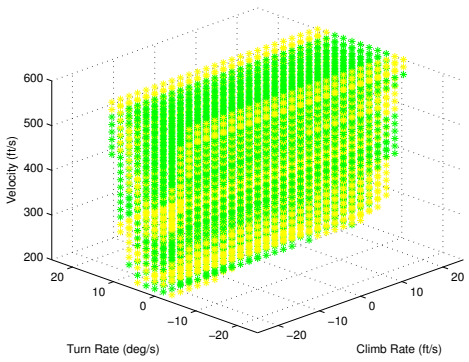
Notice that the 0° failure case differs minimally from the nominal F-16 flight envelope. These similarities occur because the rudder plays a limited role in trimming the aircraft, even during turning flight. When the rudder is failed at non-zero angles, however, the flight envelope starts to show marked degradation in overall stability. The graphs for the 15° and 30° failure cases show that when the rudder is failed with a positive deflection turns to the left remain stable, while turns to the right become unstable. This behavior is to be expected because in these non-zero failure configurations, the rudder is supplying forces and torques that promote left turns and act against right turns. To counter for these adverse effects, the aircraft requires a much larger angle-of-attack and side slip angle,



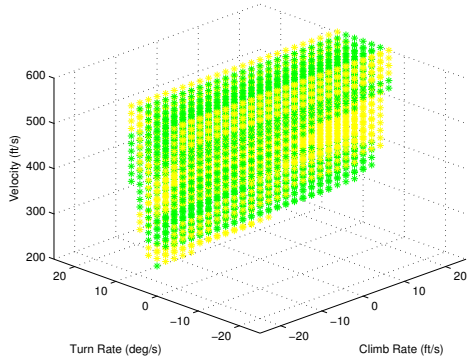
(a) 0 ft



(b) 10,000 ft

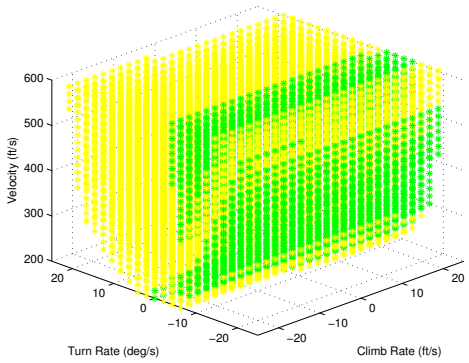


(c) 20,000 ft

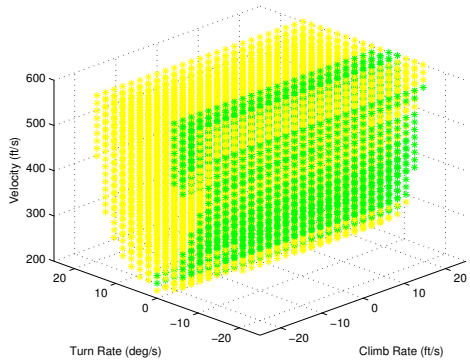


(d) 30,000 ft

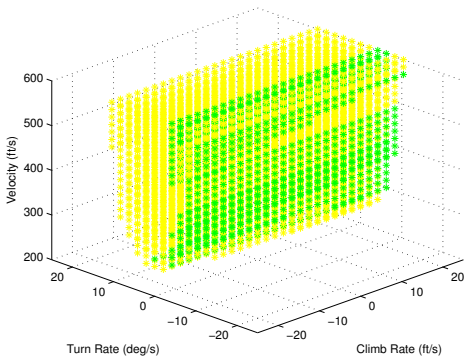
Figure 6.1: 0° Rudder Jam Flight Trim Database Slices



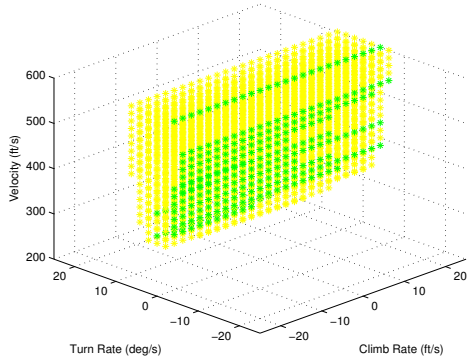
(a) 0 ft



(b) 10,000 ft

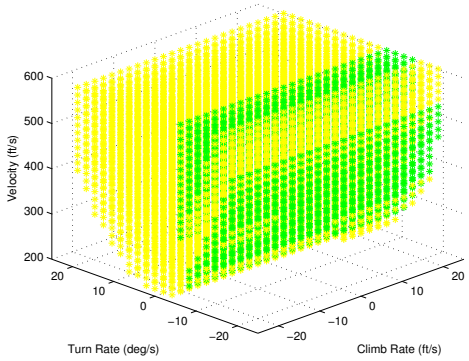


(c) 20,000 ft

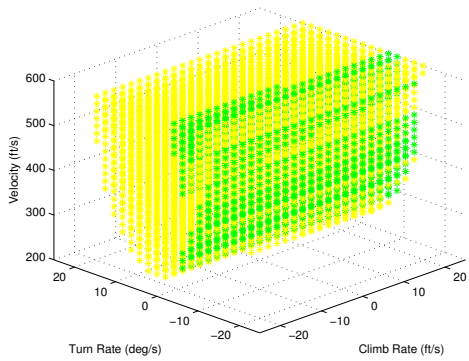


(d) 30,000 ft

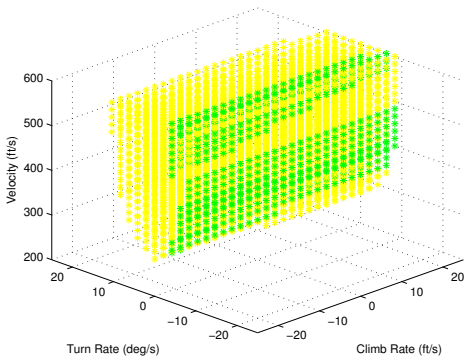
Figure 6.2: 15° Rudder Jam Flight Trim Database Slices



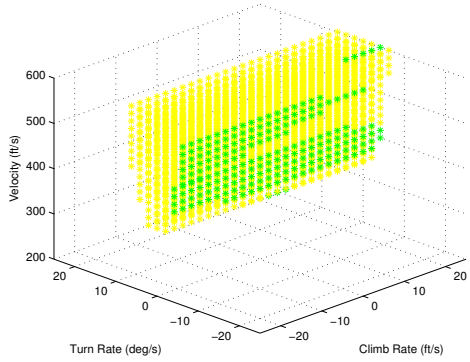
(a) 0 ft



(b) 10,000 ft



(c) 20,000 ft



(d) 30,000 ft

Figure 6.3: 30° Rudder Jam Trim Database Slices

as well as increased aileron deflections to produce the aerodynamic forces and torques necessary to balance the aircraft. The more aggressive the trim state, the more the aircraft diverges the inherent stability of symmetric flight conditions and becomes unstable.

While most of the feasible trim states in the databases are unstable, all remain controllable, even with the most severe rudder failures. Because neither the rudder or the aileron contributes significantly to trimming the aircraft under nominal flight conditions, they provide redundant lateral aircraft control (yaw and roll). In fact, of the two, the aileron is the more powerful of the two actuators. As a result, with a failed rudder, the aileron is capable of providing the necessary balancing action, as well as directing motions against the failed rudder position.

As shown in Figures 6.1-6.3, rudder failures have little effect on the range of feasible climb rates for the aircraft. Climbing flight conditions are maintained primarily through throttle adjustment and elevator deflections and only become infeasible when the thrust required to maintain that trim condition exceeds engine production limits. The minimal increase in throttle needed to compensate for rudder failures only affects the outer limits of the flight envelope, which were not used in the case studies that follow.

6.2.2 Simple 15° Rudder Jam Scenario

To test the basic operation of the planner, the first example presented here uses a relatively simple reduced trim database $\tilde{D}$. In this example, a 15° rudder jam has occured. The states in $\tilde{D}$ are all possible combinations of 5 different climb rates, 5 different turn rates, but at a single airspeed as shown in Table 6.2. Furthermore, since the volume of $\tilde{D}$—a two-dimensional plane in this case—lies

completely within its associated D', the transition map for $\tilde{D}$ is fully-connected, as mentioned in Section . $\tilde{M}$, the transition map for $\tilde{D}$, was computed for every thousand feet of altitude between sea-level and $10,000$ using the interpolated trim state controller discussed above with values of $\Delta t = 15$ and $t_c = 10$. For clarity, a graphical representation of $\tilde{D}$ and the resulting transition map is provided in Figure 6.4 where the black circles designate valid flight conditions and the green lines designate valid transitions. Finally, no additional path constraints were enforced, so each segment could be any flight condition in $\tilde{D}$, expect the permanent terminal constraint $\dot{h}_N \leq 0$.

Table 6.2: $\tilde{D}$ Values for Simple 15° Rudder Jam

Airspeed (ft/s)	Climb Rates (ft/min)	Turn Rates (deg/s)	Total Points
400	0, ±500, ±1000	0, ±3, ±6	25

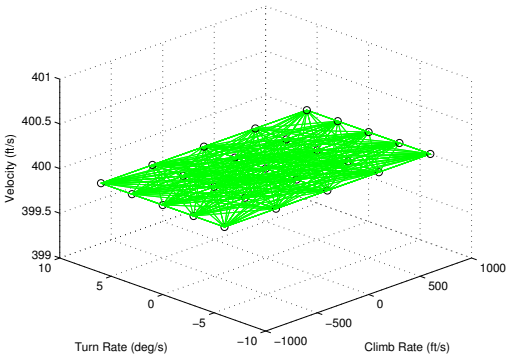
The optimal plan returned by the path planner is shown Table 6.3, where the i^{th} row summarizes the i^{th} trim segment and $i = 0$ denotes the initial aircraft flight condition. This solution was found by specifying $N = 4$ (four segments) and using the following initial aircraft location and desired landing site:

$$\mathbf{p}_0^T = [0,\ 0,\ 1000] \text{ ft}, \quad \psi_0 = 0 \text{ rad},$$

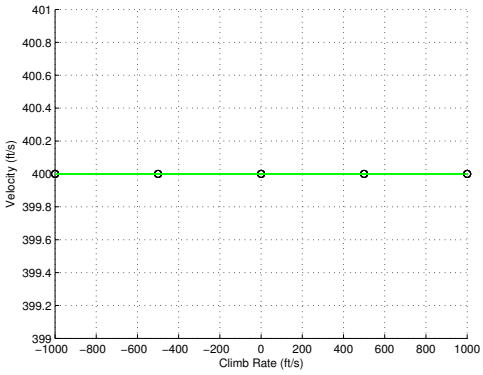
$$\mathbf{p}_{des}^T = [10000,\ 10000,\ 0] \text{ ft}, \quad \psi_{des} = 1 \text{ rad}.$$

The values in Table 6.3, along with the ramp and coast times used in $\tilde{M}$, can be used to produce the flight condition sequence shown in Figure 6.5.

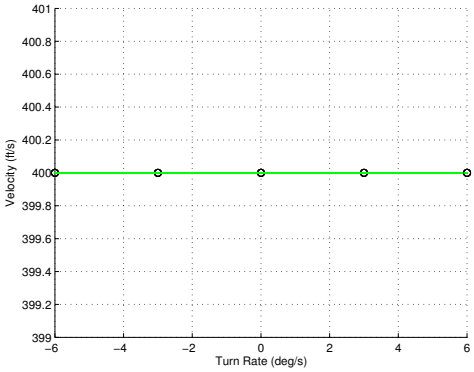
To validate this plan, another simulation of the coupled controller/aircraft system was performed. In this simulation, the aircraft was initialized at $\mathbf{p} = \mathbf{p}_0$, $\psi = \psi_0$, and $\bar{\mathbf{z}} = \bar{\mathbf{z}}^*(h_0,\ v_0,\ \dot{h}_0,\ \dot{\psi}_0)$, which is the trim state associated with the



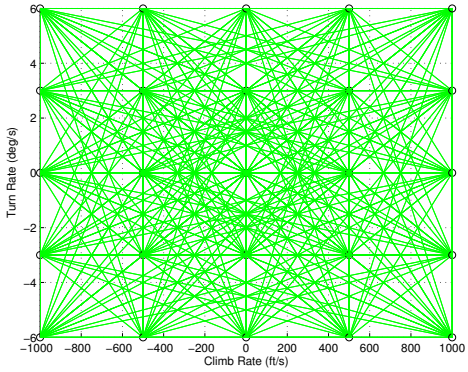
(a) 3D View



(b) x-z Projection



(c) y-z Projection



(d) x-y Projection

Figure 6.4: $\tilde{D}$ and $\tilde{M}$ for Simple 15° and 30° Rudder Jam

Table 6.3: Optimal Plan for Simple 15° Rudder Jam

i	v_i^* (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
0	400	0	0	0.00
1	400	−1000	−3	0.604
2	400	0	6	0.397
3	400	−500	3	6.351
4	400	−1000	−6	1.221
Total Plan Cost ΔT				8.573

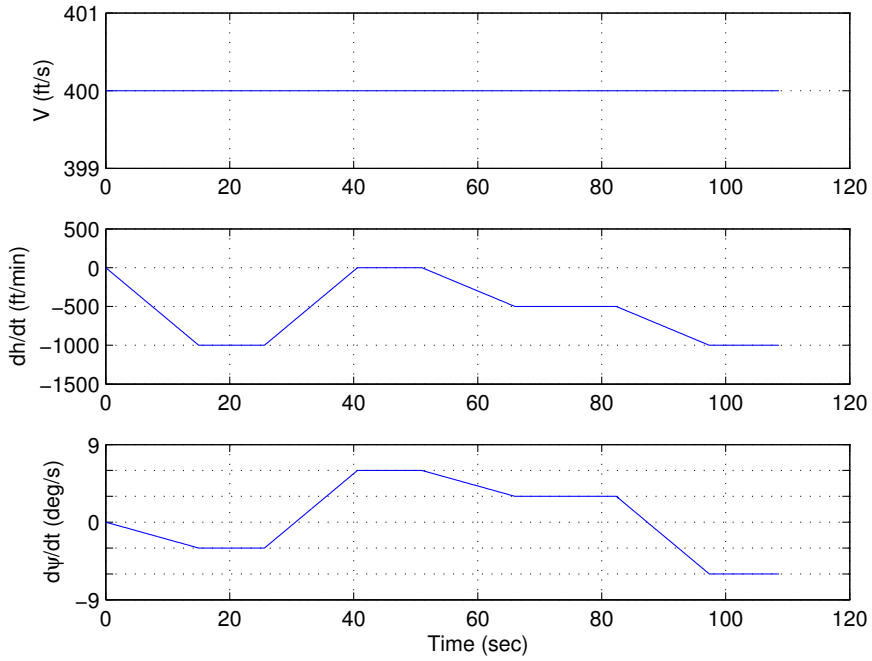
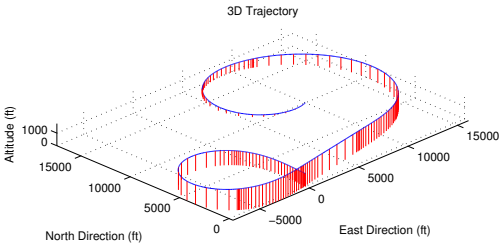
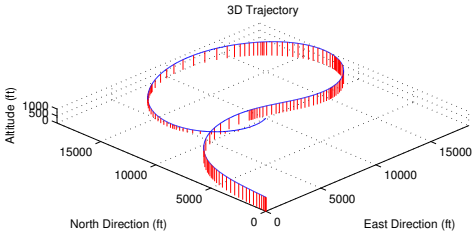


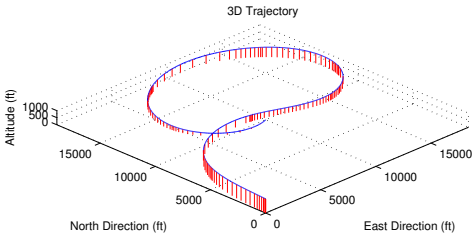
Figure 6.5: Optimal Plan for Simple 15° Rudder Jam



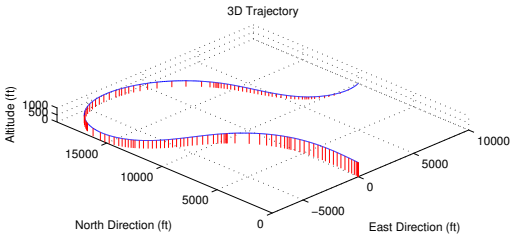
(a) First Solution



(b) Second Solution



(c) Third Solution



(d) Optimal Solution

Figure 6.6: Solution Trajectories for Simple 15° Rudder Jam

initial flight condition found in Table 6.3. Whereas it found many acceptable solutions, as the trajectory planner iterated toward an optimal solution, it updated the optimal—first—entry of the plan list L four different times. To show this iterative process, the resulting three-dimensional position trajectories for these four different plans are shown in Figure 6.6 where Figure 6.6(a) shows the first solution found and Figure 6.6(d) shows the final optimal solution. For each update, cost reduction is intuitive with respect to the total time in the air. The second solution eliminates the initial turn away from the landing site, the third solution starts descending more rapidly, and the optimal solution removes the final 360° turn.

The times at which these updates occurred, as well as the associated reduction in cost, is shown in Table 6.4, where the last row shows the information for the total planning run. Note also the speed at which the first solution is identified relative to total run time. Besides these "optimal" updates, additional solutions were found throughout the search space, as shown in Figure 6.7. This type of solution density can be expected because of the fully-connected nature of the feasible transitions. Note that the lack of solutions over the last third of the solution space correspond to the constant non-climbing terminal segment constraint.

Because $\tilde{M}$ contains only the accumulated position and heading change during a transition, an explicit planned trajectory cannot be built. Therefore, to characterize the accuracy of the trajectories produced, the planned position at the start of each segment can be compared to that of the simulated trajectory. The differences between these two sets of values are presented in Table 6.5 and show good matching between the planned and simulated trajectories. The Table 6.5 results are interesting because the individual component errors are not

Table 6.4: Solution Updates for Simple 15° Rudder Jam

Update	Time-to-Solution (sec)	Solution Cost (sec)
1	17.54	79.84
2	45.00	37.87
3	131.52	37.48
4	156.79	8.57
Total	5306.10	8.57

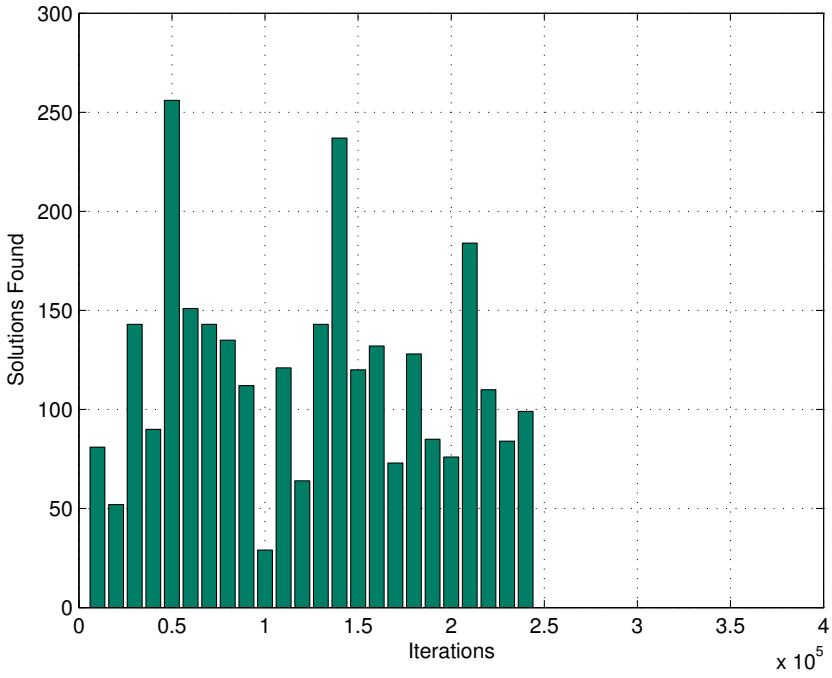


Figure 6.7: Solution Density for Simple 15° Rudder Jam

monotonically increasing over time as would be expected of an open-loop system. Because the ground track variables, particularly x, y, and ψ, are coupled, errors in each propagate through the set, making it difficult to discern where the aircraft diverges from the optimal plan. However, the total error in position and heading, represented as the norm κ_i, shows the expected increasing trend.

Table 6.5: Flight Path Errors for Simple 15° Rudder Jam

Time (sec)	x_{error} (ft)	y_{error} (ft)	h_{error} (ft)	ψ_{error} (deg)	κ
25.000	0.0805	0.0806	0.0003	0.0000	0.1139
50.604	−0.4936	−0.6002	0.0013	0.0000	0.7771
76.001	0.1584	−0.5362	0.0018	0.0000	0.5591
107.352	−0.3271	−1.2917	0.0091	0.0000	1.3325
108.573	−0.2484	−1.3115	0.0089	0.0000	1.3349

To gain physical understanding of the optimal plan exectuion, Figure 6.8 shows the flight condition over time versus the planned flight conditions of Figure 6.5, as well as the errors between them. To aide in discerning separate trim and transition segments, the black dots denote trim segment beginnings (or transition ends) and red dots transition beginnings (or trim segment ends). Note that motion nonlinearities (non-kinematic motion) are constrained to the transition segments due to the use of kinematic trim segments; the flight condition error over trim segments is defined to be zero.

As alluded to above, in order for the aircraft to maintain a specific trim condition, it must assume a very aggressive stance with respect to the local wind. Figure 6.9 clearly shows this phenomenon by presenting the required orientation

for the optimal flight plan as well as the differences between the simulated and ideal trim values. The first interesting thing to note is the initial orientation: the specified initial flight condition was straight and level flight but requires non-zero bank and side slip angles. This "slipping" configuration is required to balance the rudder failure effects: the positive rudder failure produces a negative yawing moment, pushing the nose to the left, which the positive side slip cancels by introducing a non-zero side-force that creates a positive yawing moment. The rudder failure also introduces a positive rolling moment, producing the non-zero bank angle seen in the response. Since the F-16 produces negative rolling moments for positive side slip angles, that is, the aircraft has a negative C_{L_β} stability derivative over the range of β for which the F-16 model is valid [31], the slipping motion also cancels the rolling motion produced by the positive rudder failure. This balancing continues throughout the trajectory with coupled transients in ϕ and β.

Similarly, the angle-of-attack and pitch angles also show large transients. Particularly interesting is the initial divergent response of α from desired reference values. The unsteady response in climb and turn rate, and the subsequent lag in compensation, is a by-product of the controller mismatch discussed in Section 6.1; α varies quadratically with climb rate. As the controller tries to minimize errors in $\dot{h}$, it introduces errors in α and *vice versa*.

The orientation variables also show coupling to the commanded control input as seen in Figure 6.10. Notice the pronounced coupling between the angle-of-attack lag and elevator deflections. Figure 6.10 also shows the additional actuator deflections needed to dampen the undesirable natural aircraft transients, most notably the phugoid and spiral modes discussed in Section 4.2.1. Note that

these deviations from the necessary trim settings are minimal and closely mimic nominal pilot-issued actuator commands.

Also note the time spent in the trim segments is far less than time spent in the transitions, which is a result of the closeness of the desired landing site. Recall the planning algorithm can only optimize over trim segment durations and that transition durations have been fixed off-line. In this case, the planner determined the most optimal course of action is to "ride" the transients, resulting in a mostly dynamic trajectory.

6.2.3 Simple 30° Rudder Jam Scenario

For the next test, a 30° rudder jam was simulated. The $\tilde{D}$ shown in Figure 6.4 and Table 6.2 was used with the same time and altitude values in compiling $\tilde{M}$. The optimal plan produced by the planner for this case is summarized in Table 6.6 and shown graphically in Figure 6.11.

To verify the planner iteration, Figure 6.12 shows the three-dimensional trajectory for the first solution found (Figure 6.12(a)), the first update to L (Figure 6.12(b)), the third update (Figure 6.12(c)), and the final optimal plan (Figure 6.12(d)). Note that this iterative adjustment matches that of the 15° rudder jam, though the final solutions differ. The optimal solution here is to climb first, then descend. These differences in plans affect the final ΔT, which is over three seconds higher here. The time-to-solution and costs for these updates are provided in Table 6.7 and the solution density graph is shown in Figure 6.13. Both show trends similar to the 15° rudder jam case.

A comparison of the simulated and planned solutions is shown in Table 6.8. This solution has similar traits to the 15° rudder jam: increasing error magnitude

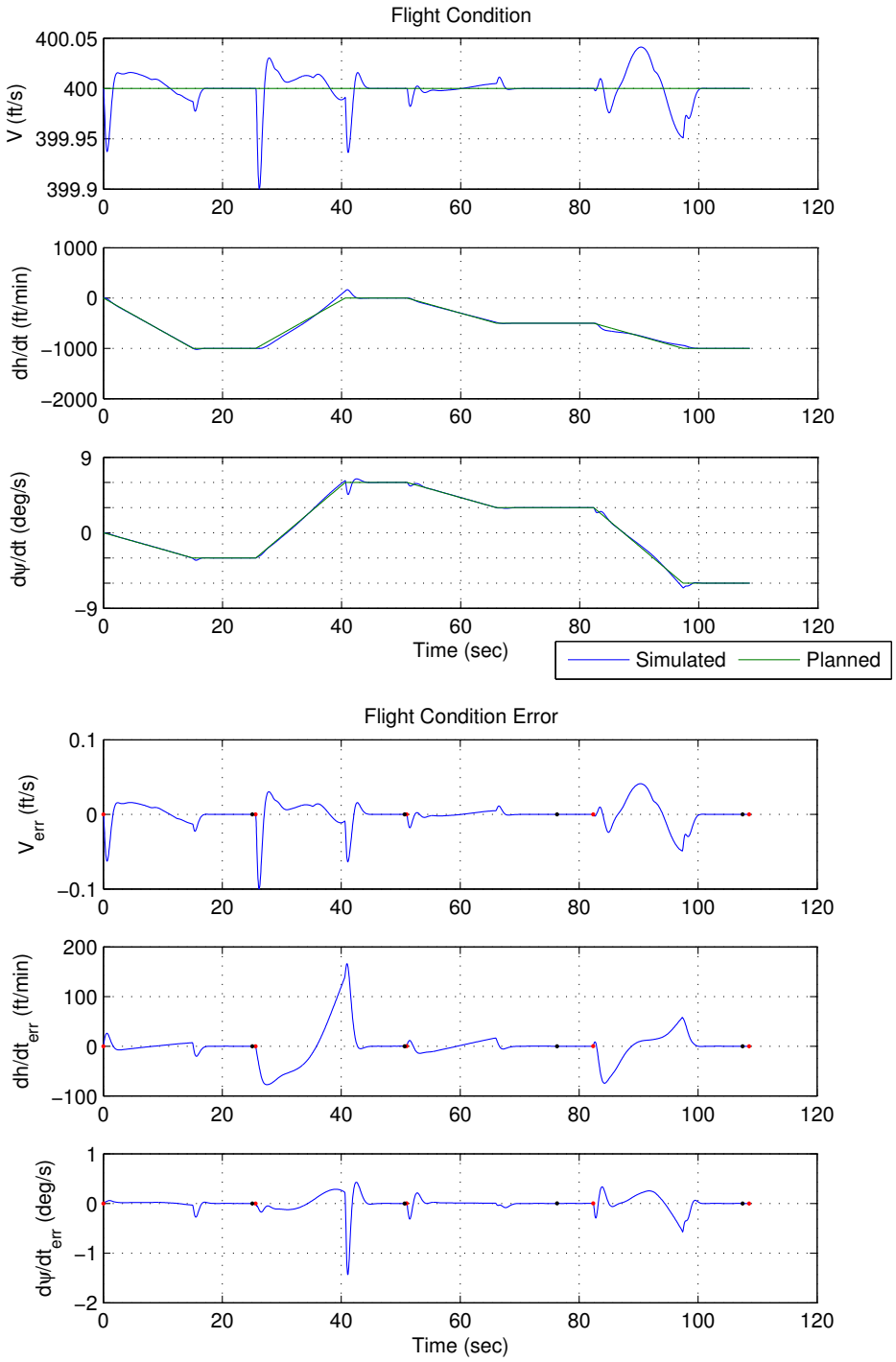


Figure 6.8: Flight Condition Information for Simple 15° Rudder Jam

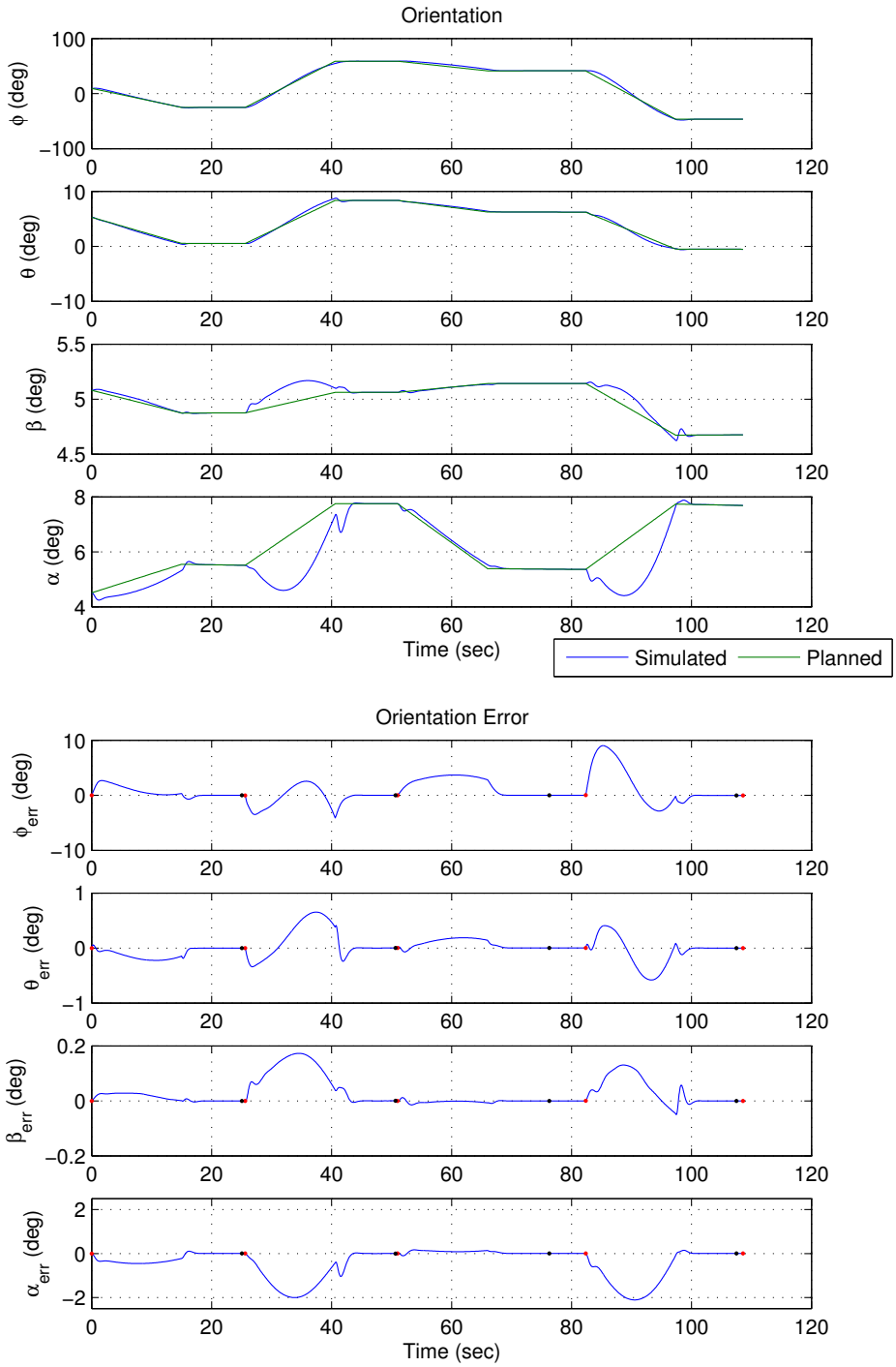


Figure 6.9: Orientation Information for Simple 15° Rudder Jam

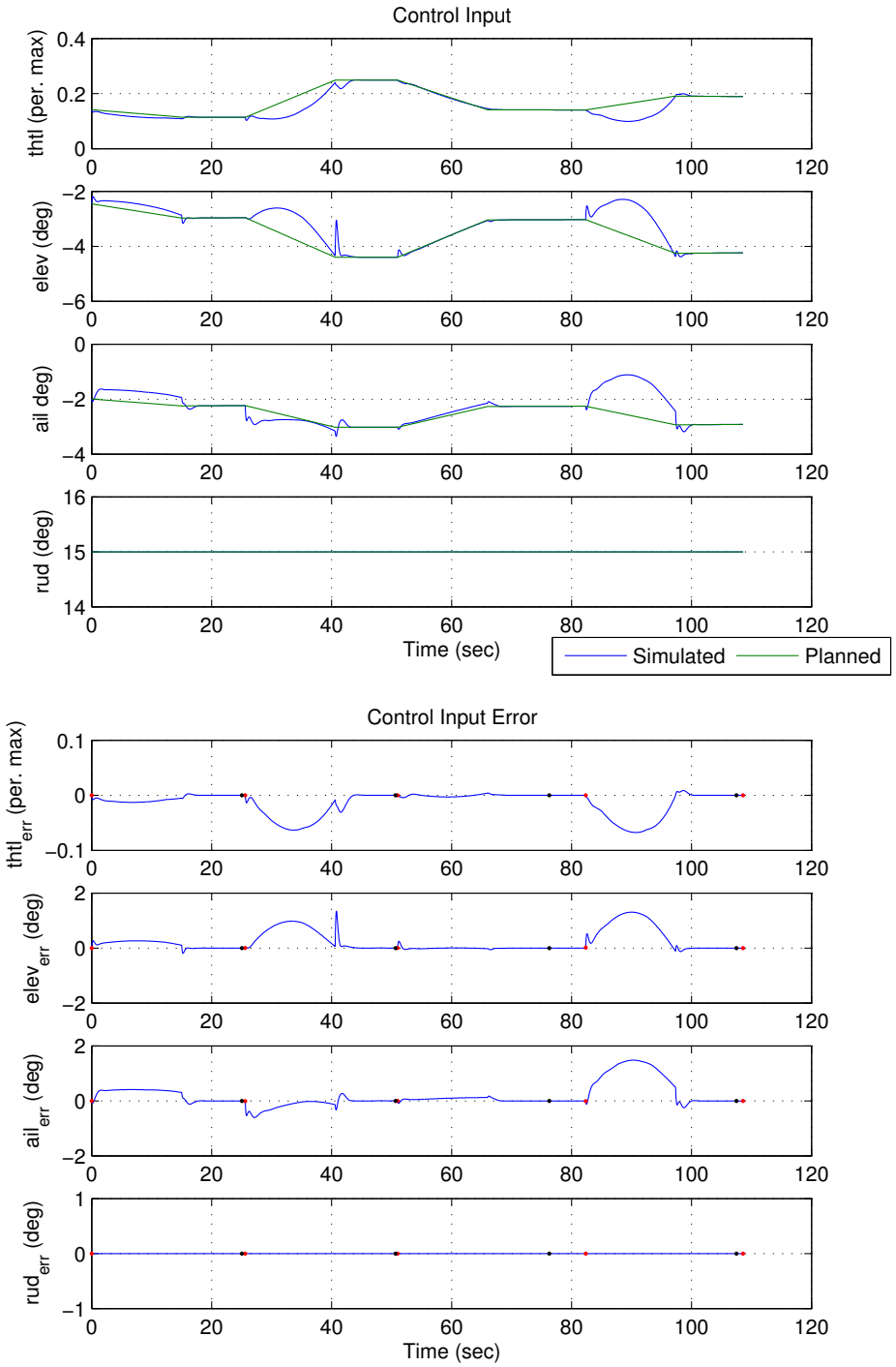


Figure 6.10: Controller Information for Simple 15° Rudder Jam

Table 6.6: Optimal Plan for Simple 30° Rudder Jam

i	v_i^* (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
0	400	0	0	0.00
1	400	500	−3	0.189
2	400	−1000	6	1.857
3	400	−1000	3	2.350
4	400	−1000	−6	0.889
Total Plan Cost ΔT				5.287

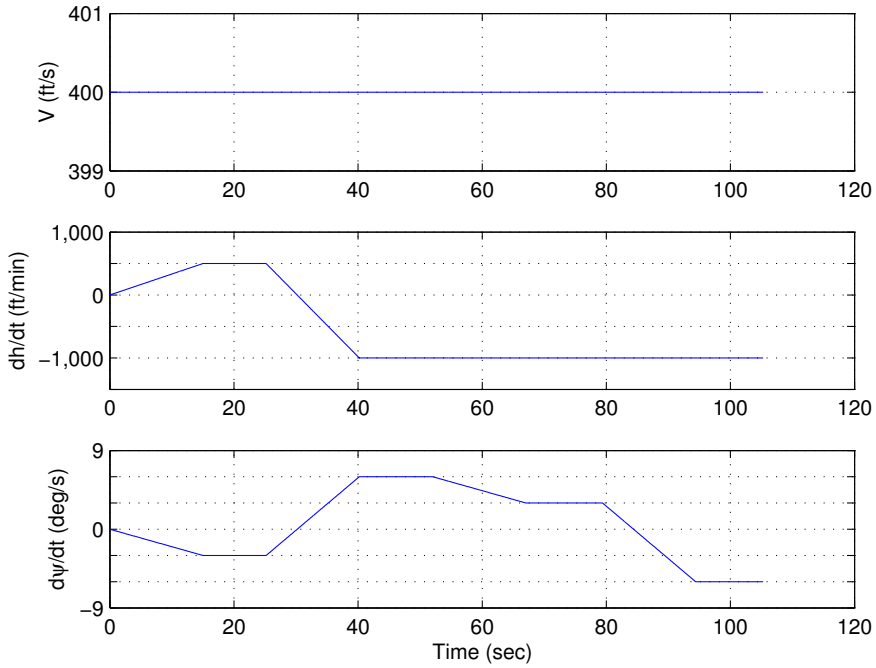
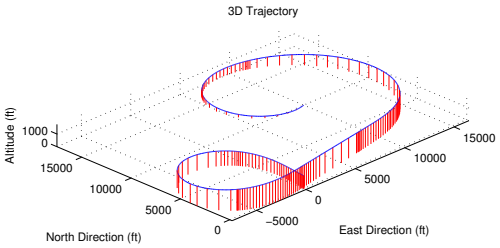
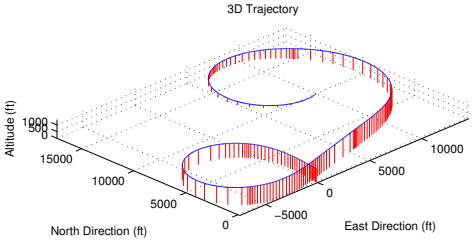


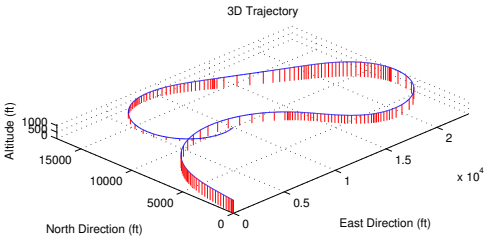
Figure 6.11: Optimal Plan for Simple 30° Rudder Jam



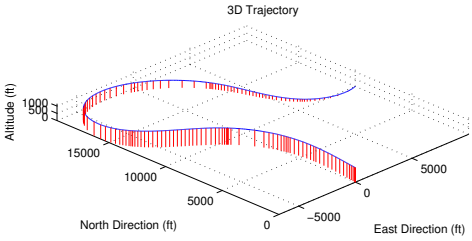
(a) First Solution



(b) Second Solution



(c) Fourth Solution



(d) Optimal Solution

Figure 6.12: Solution Trajectories for Simple 30° Rudder Jam

Table 6.7: Solution Updates for Simple 30° Rudder Jam

Update	Time-to-Solution (sec)	Solution Cost (sec)
1	24.89	80.81
2	28.37	70.63
3	51.69	68.39
4	59.51	66.40
5	71.53	5.28
Total	6489.03	5.28

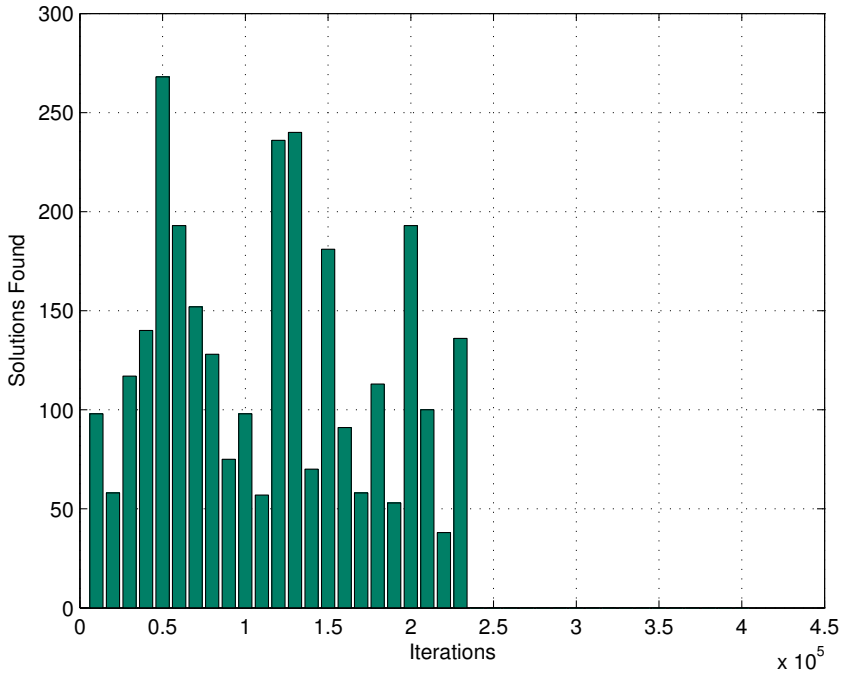


Figure 6.13: Solution Density for Simple 30° Rudder Jam

κ and close to perfect matching of heading ψ and altitude h. The evolution of the flight condition is provided in Figure 6.14. The similarity in the failure scenario between the 15° and 30° rudder jams yields similar transients in the flight condition during transitions. The 30° rudder jam also leads to aggressive stances to maintain trimmed flight as can be seen in Figure 6.15, most notably in large side slip angles, almost 9°. The required control response, shown in Figure 6.16, also experiences larger deviations than required in the 15° jam scenario.

Table 6.8: Flight Path Errors for Simple 30° Rudder Jam

Time (sec)	x_{error} (ft)	y_{error} (ft)	h_{error} (ft)	ψ_{error} (deg)	κ
25.000	0.0007	0.0110	−0.0003	0.0000	0.0131
50.189	0.0025	−0.3321	−0.0008	0.0000	0.3331
77.046	0.2985	−0.0896	0.0029	0.0000	0.3117
104.396	0.0664	0.6867	0.0090	0.0000	0.6899
105.285	0.1753	0.6554	0.0089	0.0000	0.6785

6.2.4 Complex 15° Rudder Jam Scenario

In this final example in the rudder case study, a more complicated $\tilde{D}$ will be examined consisting of feasible flight conditions from the 15° trim database with values selected at now three different airspeeds, climb rates, and turn rates provided in Table 6.9 and shown in Figure 6.17. This $\tilde{D}$ structure now includes transitions between airspeeds, as well as climb and turn rates. However, such transitions require increased throttle response to reach the terminal flight condition in the alloted time. Therefore, when compiling $\tilde{M}$ with the values $\Delta t = 20$

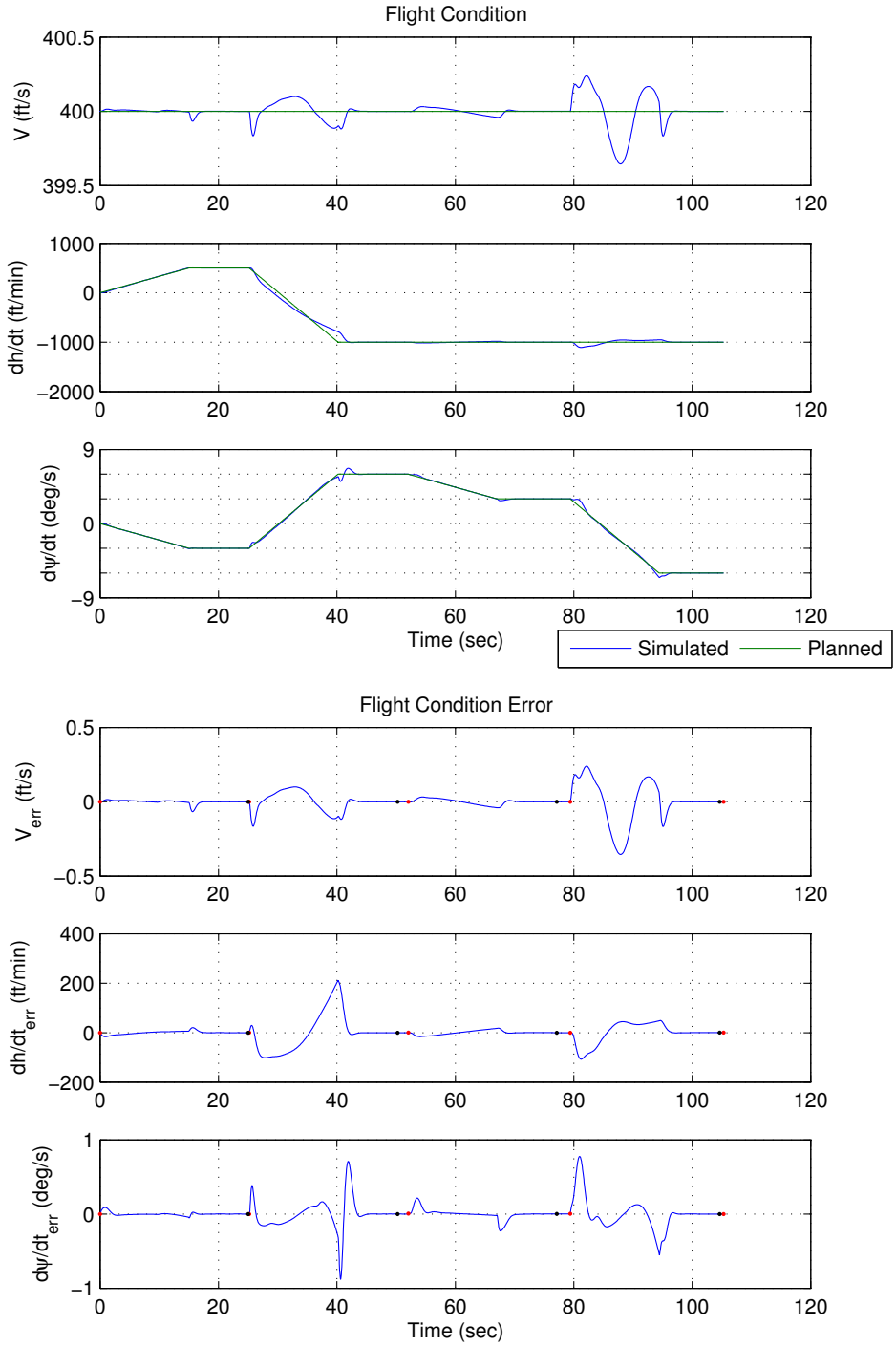


Figure 6.14: Flight Condition Information for Simple 30° Rudder Jam

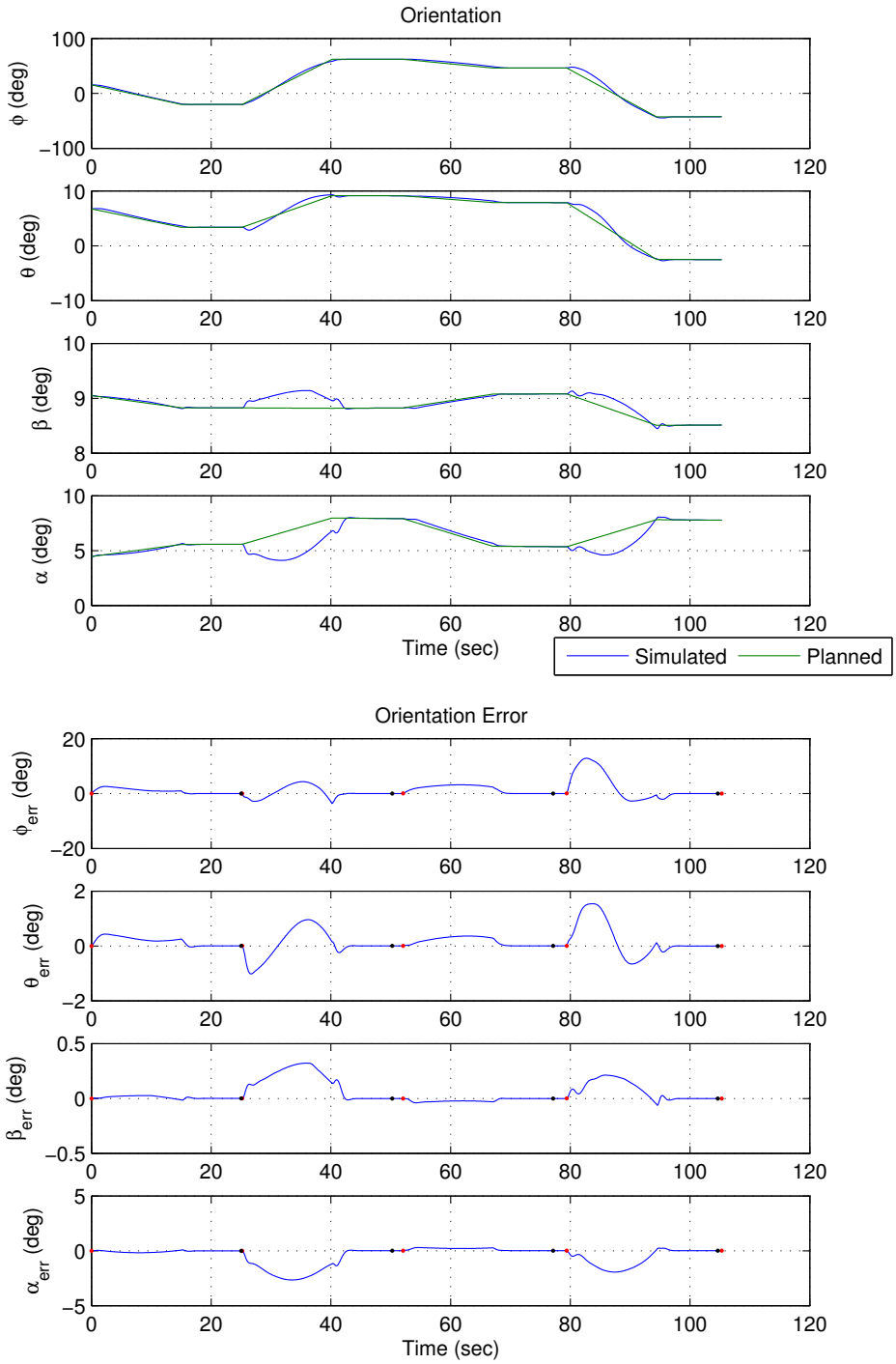


Figure 6.15: Orientation Information for Simple 30° Rudder Jam

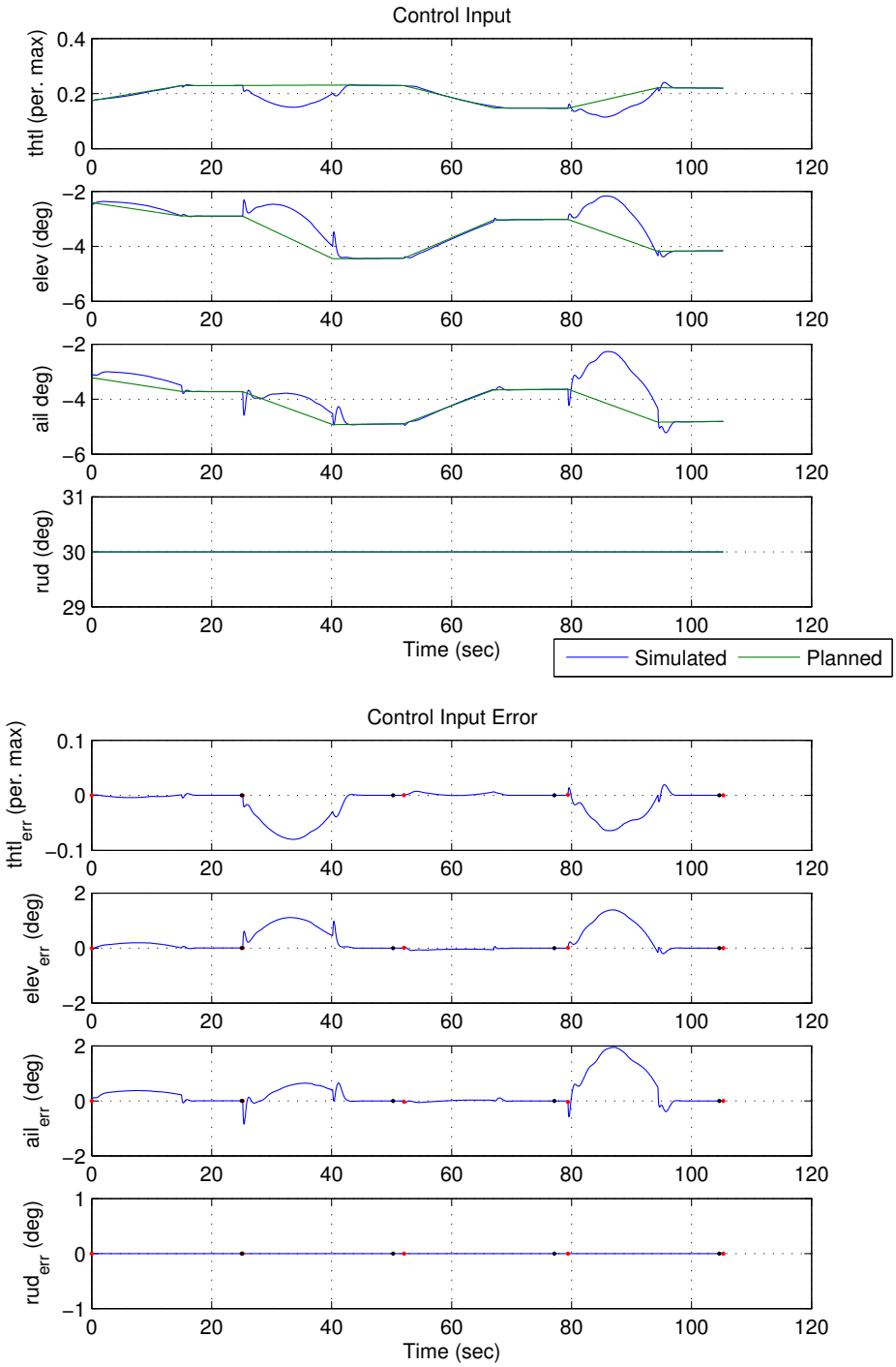


Figure 6.16: Controller Information for Simple 30° Rudder Jam

sec and $t_c = 10$ sec, those transitions of more than a ± 50 ft/sec change in airspeed when observed to saturate the throttle setting. Those transitions that are feasible are denoted as the links in Figure 6.17.

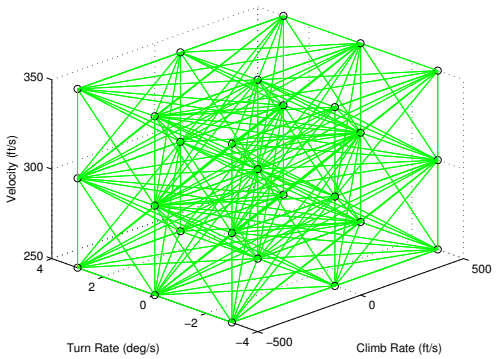
Table 6.9: $\tilde{D}$ Values for Complex 15° Rudder Jam

Airspeed (ft/s)	Climb Rates (ft/min)	Turn Rates (deg/s)	Points
250	0, ± 500	0, ± 3	9
300	0, ± 500	0, ± 3	9
350	0, ± 500	0, ± 3	9
Total Points			27

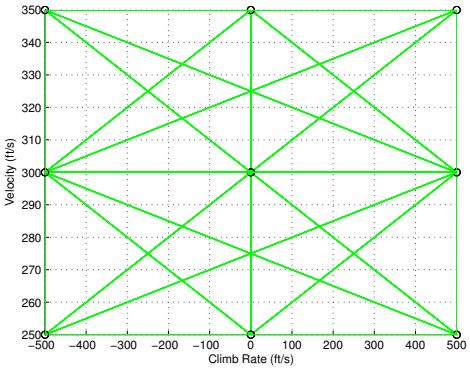
In this example, additional constraints were also placed on the sequencing of valid solutions requiring non-increasing airspeed seqeunces. Also, a terminal constraint requiring $v_N = 250$ ft/sec was specified. Such constraints more accurately model typical landing scenarios where the pilot slows the aircraft on approach, with the final airspeed slightly above stall speed.

The optimal solution found, using the same initial aircraft location, desired landing site, and number of segments as above, is summarized in Table 6.18 and shown graphically in Figure 6.18. The effect of limiting the range of feasible climb and turn rates in $\tilde{D}$ is clearly seen in the increased optimal cost than compared to the simpler $\tilde{D}$ used in Section 6.2.2. Recall that, using this simple $\tilde{D}$, the planner found that high turn and climb rates gave an optimal result of about 8 seconds. By eliminating these high turn and climb rates, the planner must find a less aggressivly constrained solution that will likely have higher cost ΔT.

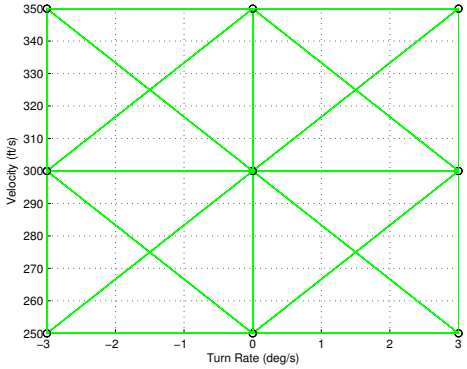
Coincidently, in this example, the optimal solution was also the first solution



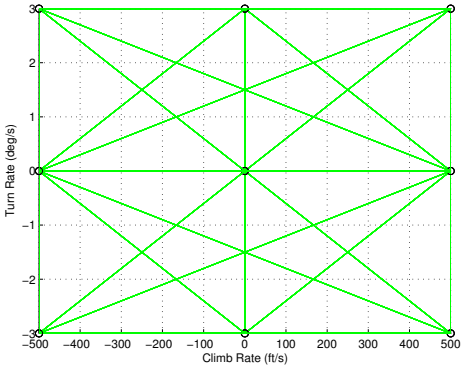
(a) 3D View



(b) x-z Projection



(c) y-z Projection



(d) x-y Projection

Figure 6.17: $\tilde{D}$ and $\tilde{M}$ for Complex 15° Rudder Jam

Table 6.10: Optimal Plan for Complex 15° Rudder Jam

i	v_i^* (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
0	350	0	0	0.00
1	300	−500	−3	8.812
2	250	0	3	48.835
3	250	−500	0	30.265
4	250	−500	−3	0.924
Total Plan Cost ΔT				88.836

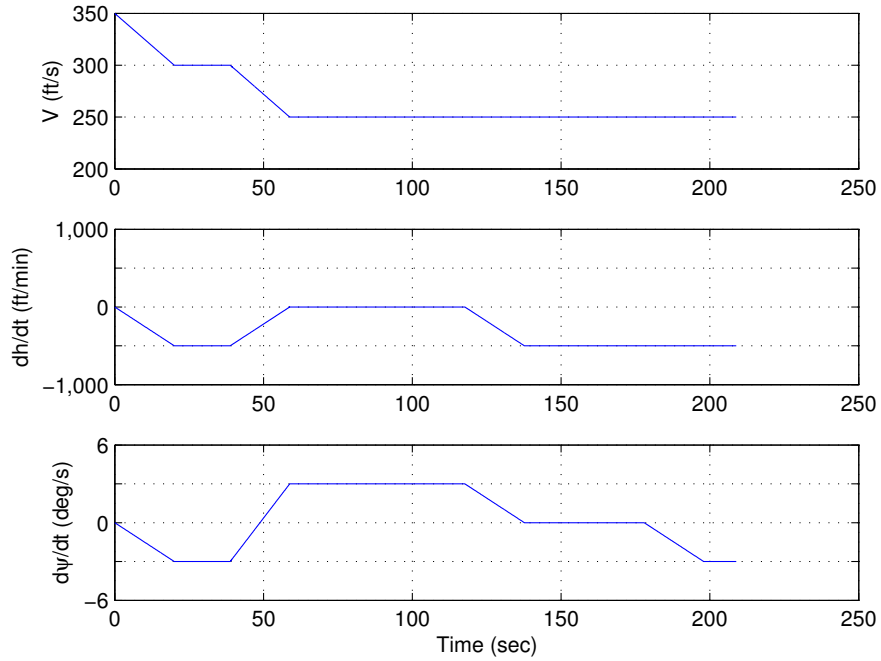


Figure 6.18: Optimal Plan for Complex 15° Rudder Jam

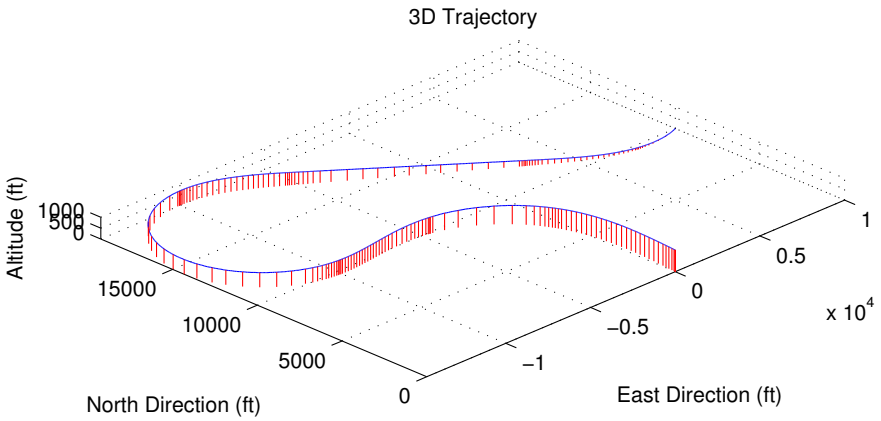


Figure 6.19: Optimal Trajectory for Complex 15° Rudder Jam

found. The three-dimensional trajectory for this solution is shown in Figure 6.19. A summary of the optimal solution, and search properties, is provided in Table 6.12. Note that the partially-connected transition map significantly reduced the total computation time, even though the number of iterations increased. The sparser transition map, along with the additional sequence constraints, also effected the solution density shown in Figure 6.20. The error in the simulated trajectory shown in Figure 6.19 is summarized in Table 6.11.

Table 6.11: Flight Path Errors for Complex 15° Rudder Jam

Time (sec)	x_{error} (ft)	y_{error} (ft)	h_{error} (ft)	ψ_{error} (deg)	κ
30.000	0.0559	0.0441	−0.0001	0.0000	0.0712
68.812	0.0681	−0.1378	−0.0009	0.0000	0.1537
147.647	−0.1703	−0.3821	0.0065	0.0000	0.4184
207.912	−0.2448	−0.3385	0.0073	0.0000	0.4178
208.836	−0.1265	−0.3840	0.0073	0.0000	0.4044

The flight condition evolution for the optimal solution is shown in Figure 6.21 and shows good matching with minimal deviations. The existence longer duration trim segments, not seen in the examples above, illustrates the excellent performance of the controller in tracking a desired trimmed flight condition. The longer trim segments also accentuate the nonlinearities confined to the transitions. Figure 6.21 also shows the controller's ability to track different airspeeds. Not only does the controller perfectly track each commanded change in airspeed, but switching between airspeeds does not adversely effect controller performance in tracking varied climb and turn rates.

Table 6.12: Solution Updates for Complex 15° Rudder Jam

Update	Time-to-Solution (sec)	Solution Cost (sec)
1	10.15	88.84
Total	1596.61	88.84

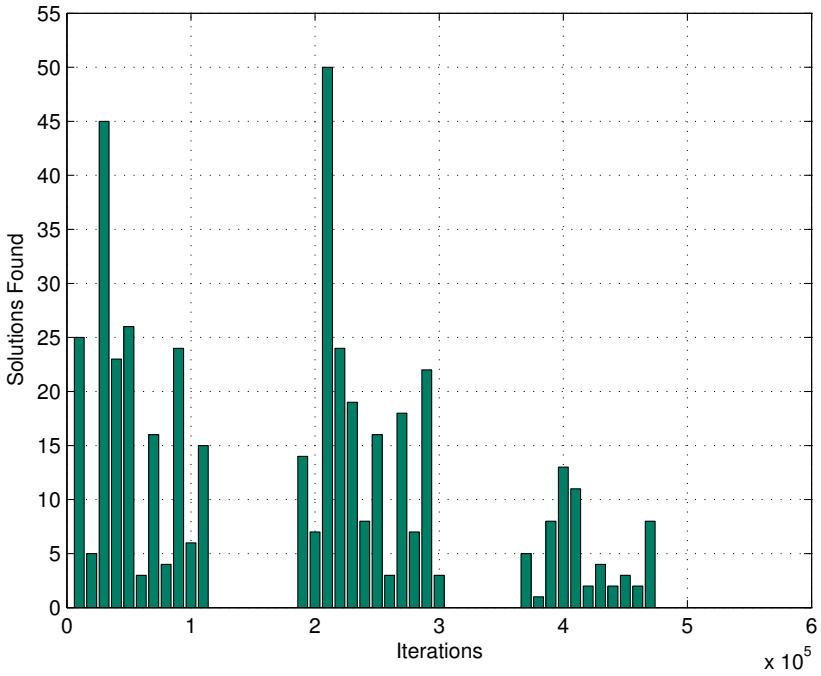


Figure 6.20: Solution Density for Complex 15° Rudder Jam

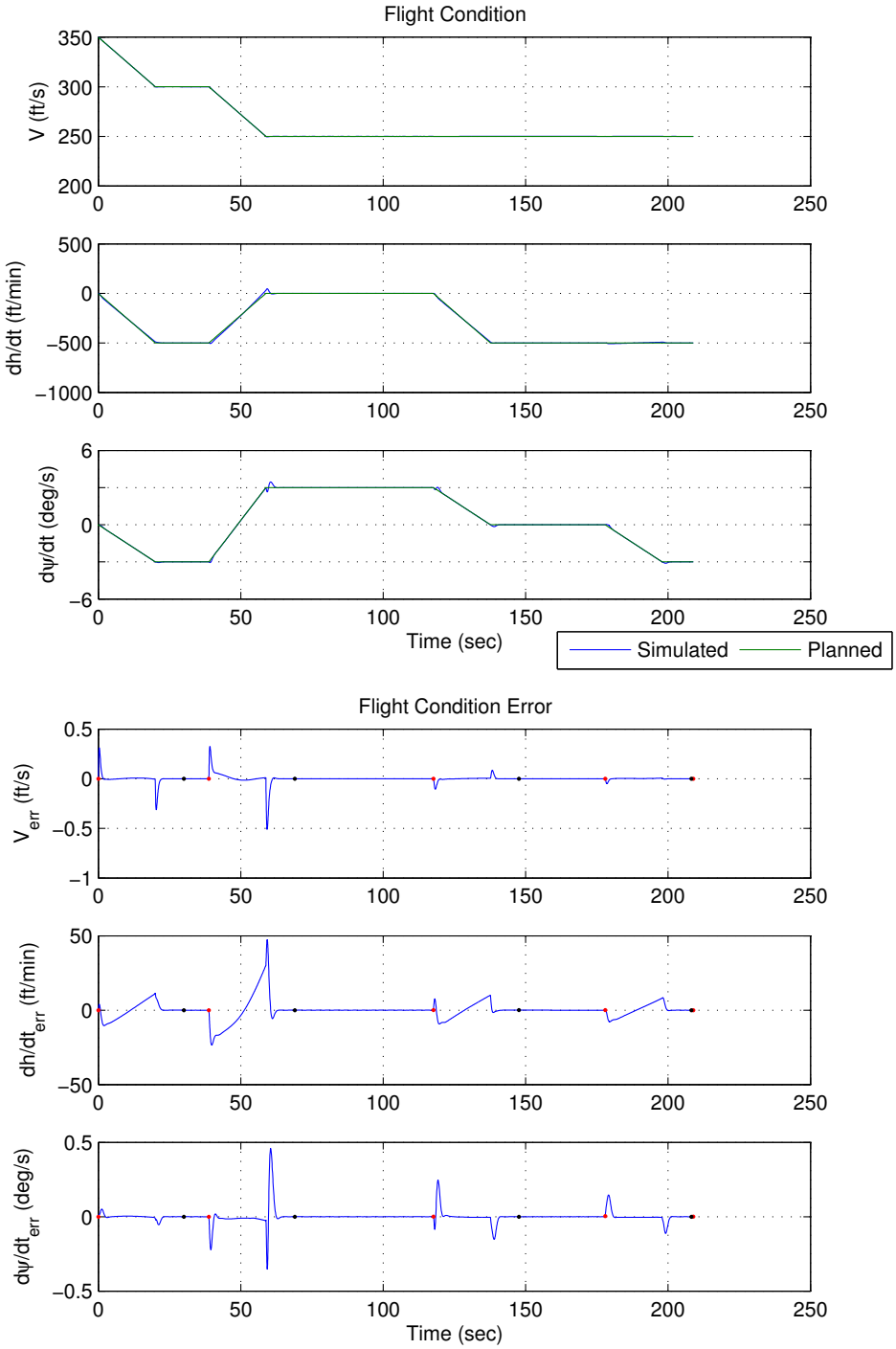


Figure 6.21: Flight Condition Information for Complex 15° Rudder Jam

The effects of varying the airspeed along the trajectory can be better seen by examining the required orientation (Figure 6.22) and required control input (Figure 6.23). Slowing the aircraft reduces the effectiveness of the aerodynamic surfaces on the vehicle, including the vehicle body itself. To retain sufficient lift, the aircraft must increase the angle-of-attack to "use" more of the air flowing over its body. These high angle-of-attack maneuvers are accentuated in segments two, three, and four.

Additionally, commanding these slower airspeeds is accomplished primarily through varying the throttle setting. Figure 6.23 shows these "step-like" changes in throttle used to slow the vehicle. The decreased throttle setting, coupled with a pitched-up configuration, emulates the standard method used by pilots to manually perform the same decelerating maneuver. The required control response also shows that this deceleration requires minimal additional control authority due to the natural tendency of the aircraft, under low thrust, to slow. Once the aircraft has attained the desired airspeed, though, a sharp deflection of the elevator is needed to counter the sharp change in throttle setting. For the non-decelerating segments (2, 3, and 4), the need for minimal corrective control input reflects the smaller flight condition transients seen in Figure 6.21, which can be seen as a result of both smaller transitions and slower airspeeds.

6.3 Aileron Failure

6.3.1 Trim Database Calculation

For the next test series, trim databses calculated for aileron jams at 0°, 5°, and 10°, with the results shown in Figures 6.24-6.26. The same values of $\epsilon_{trim} = 10^{-7}$, $\epsilon_{stab} = 10^{-3}$, and $\epsilon_{ctrb} = 10^{-12}$ were used to grade feasibility, stability, and

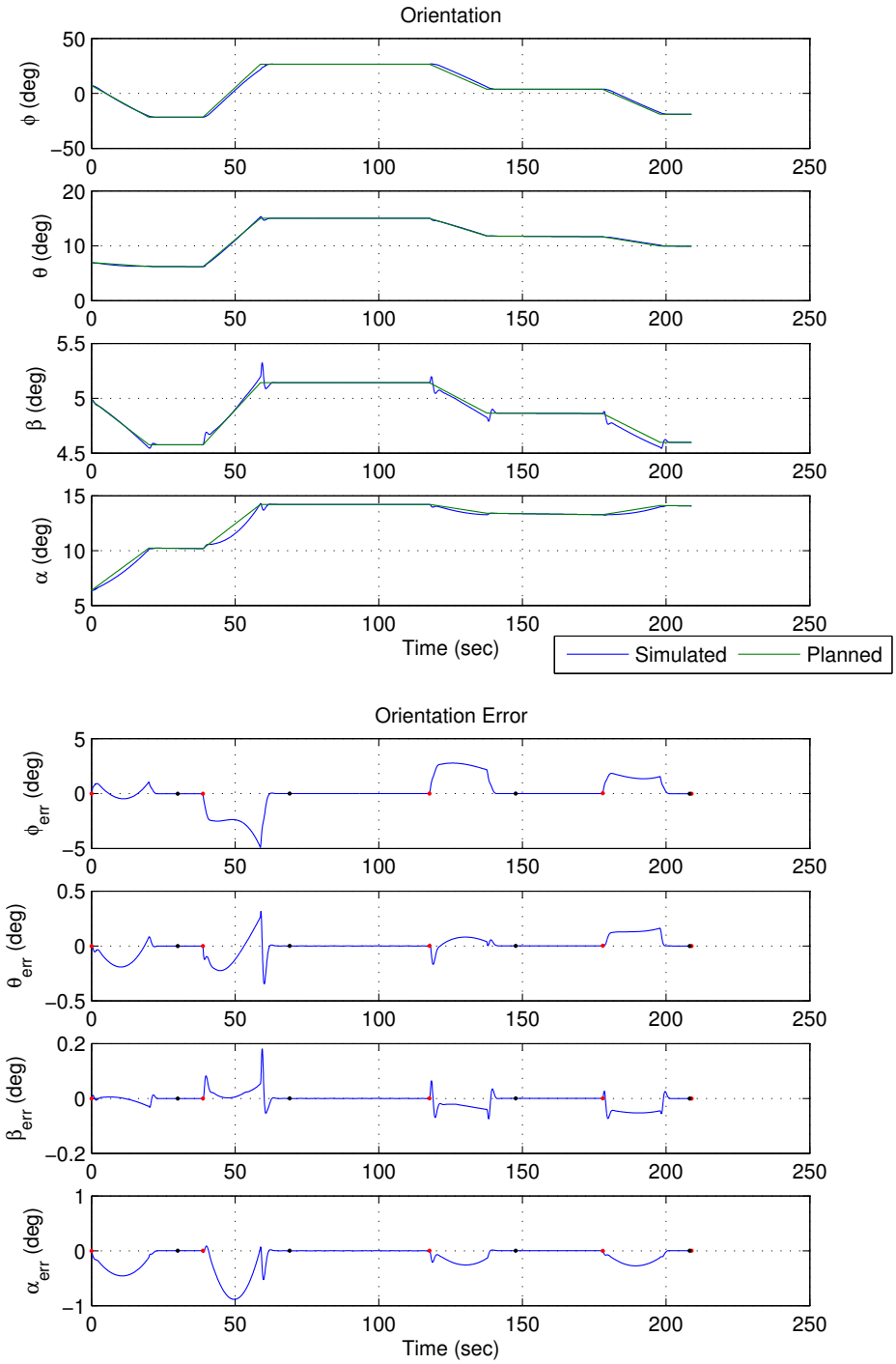


Figure 6.22: Orientation Information for Complex 15° Rudder Jam

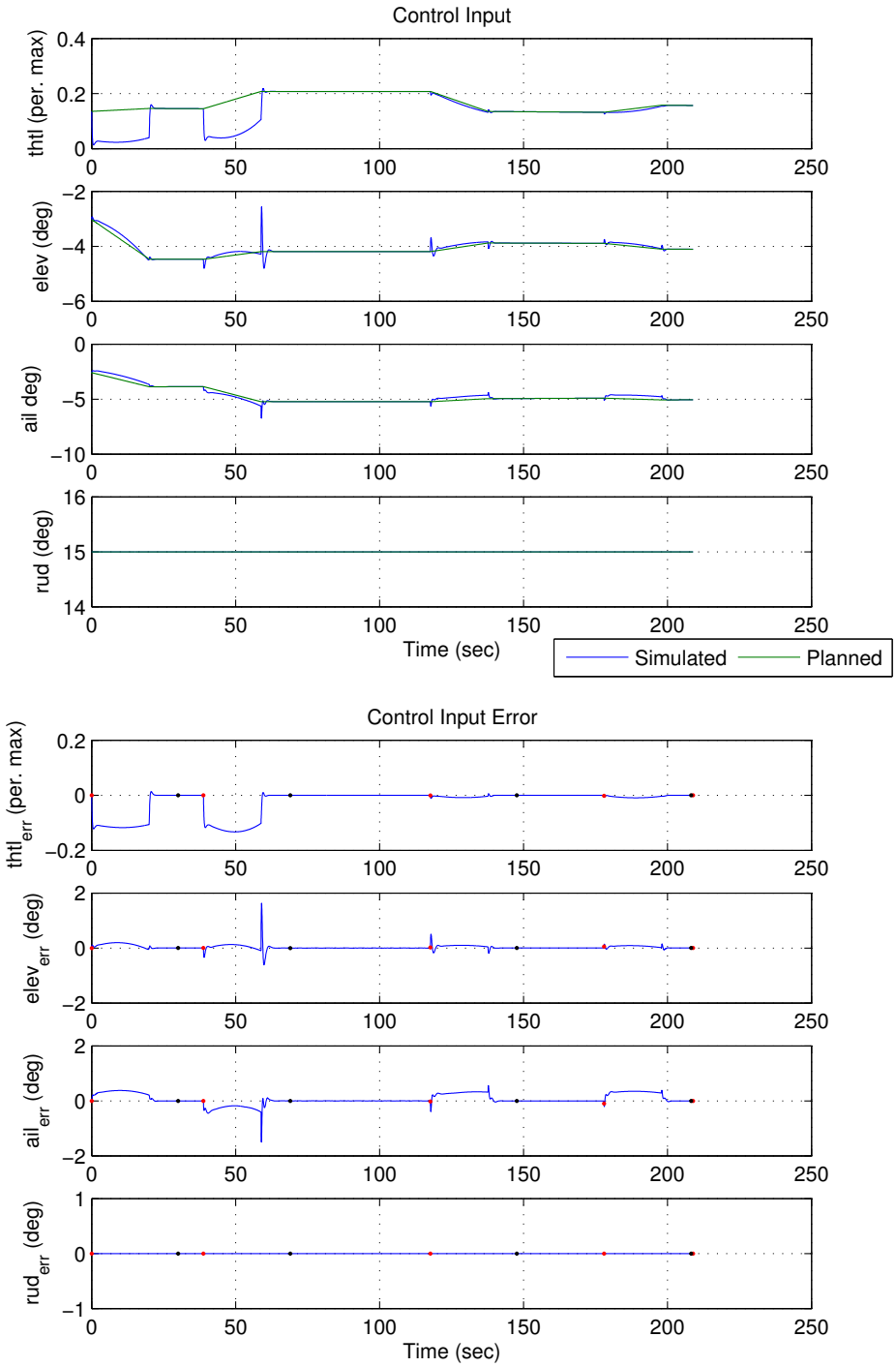
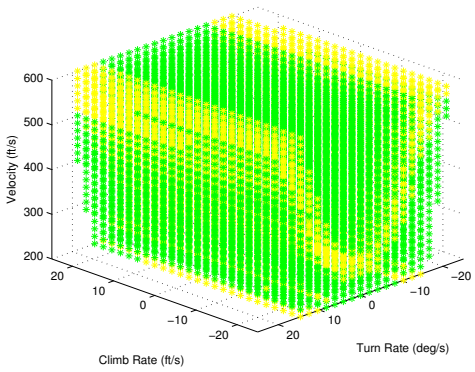
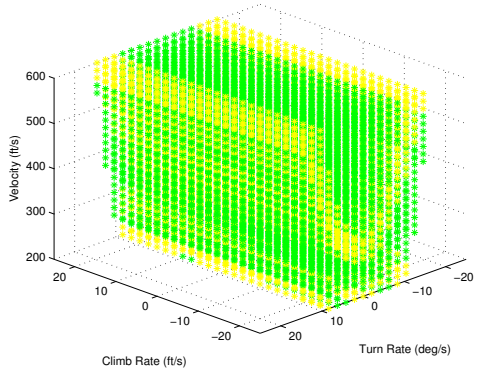


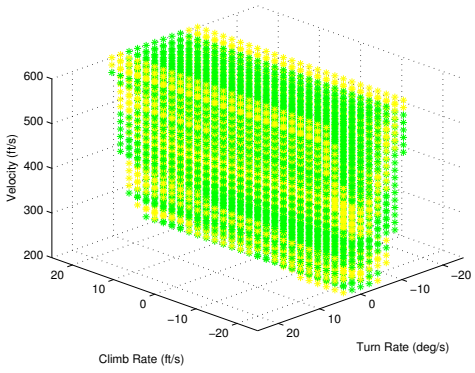
Figure 6.23: Controller Information for Complex 15° Rudder Jam



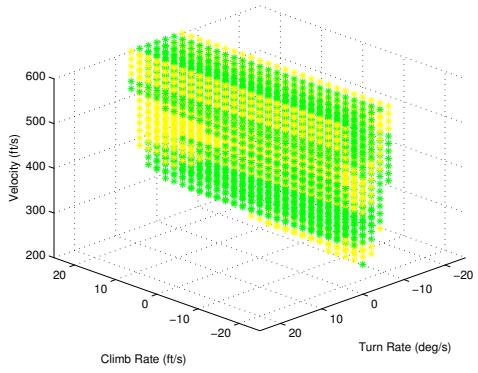
(a) 0 ft



(b) 10,000 ft

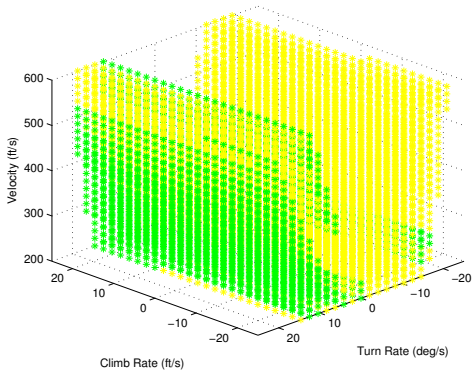


(c) 20,000 ft

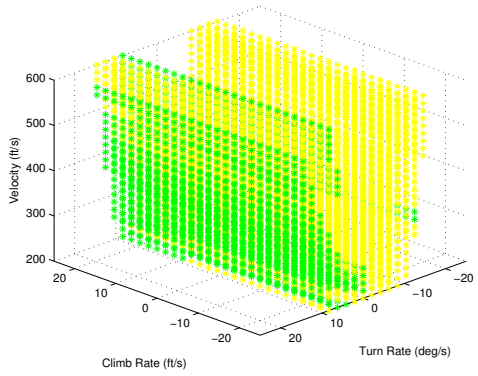


(d) 30,000 ft

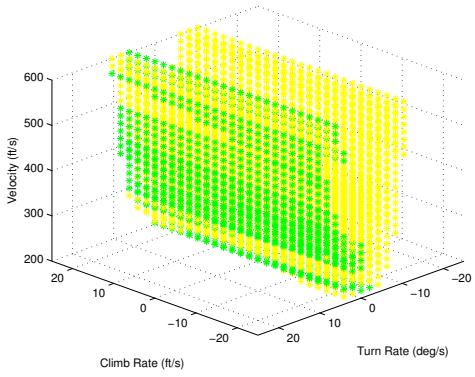
Figure 6.24: 0° Aileron Jam Trim Database Slices



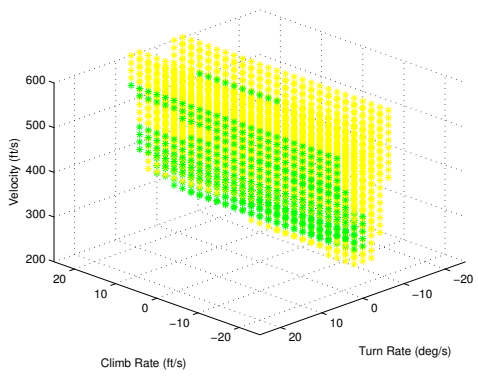
(a) 0 ft



(b) 10,000 ft

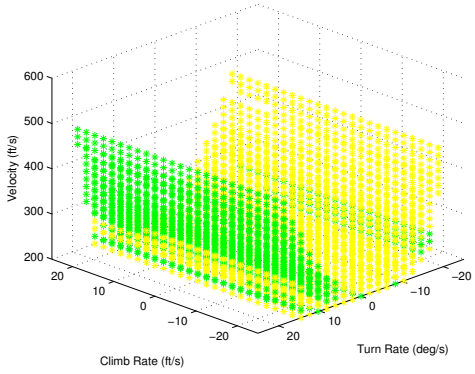


(c) 20,000 ft

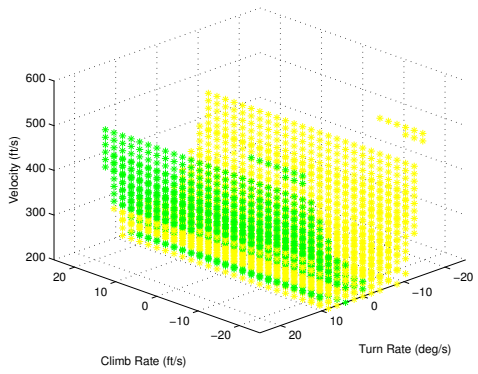


(d) 30,000 ft

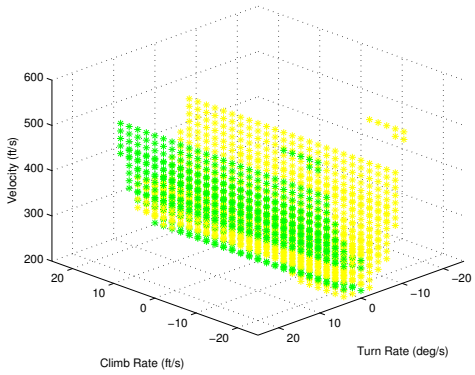
Figure 6.25: 5° Aileron Jam Trim Database Slices



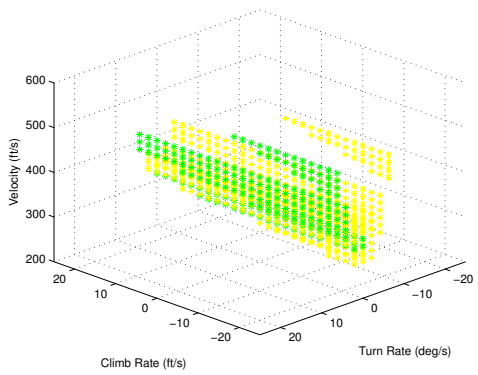
(a) 0 ft



(b) 10,000 ft



(c) 20,000 ft



(d) 30,000 ft

Figure 6.26: 10° Aileron Jam Trim Database Slices

controllability of the aircraft at each sampled point (see Table 6.1).

Similar to the rudder failure case, the 0° failure shows no alteration from the nominal, unfailed configuration due to the limited contribution of the aileron in nominally trimming the aircraft. The nonzero failure cases, however, show sizeable flight limitations, particularly for straight or near-straight flight. The definitive U-shaped flight envelope given an aileron failure reflects the complex relationship between aileron effectiveness and atmospheric flight conditions. As with all aerodynamic control surfaces, aileron effectiveness is dictated by the dynamic pressure acting on the vehicle, which, in turn, is a function of the vehicle airspeed, as well as density (or altitude). The faster the vehicle flies, or the closer to sea-level it flies, the more effective the aileron becomes. As a result, countering the effects of a failed aileron in these same flight regions becomes difficult, and in some cases, impossible. At high velocities there exist regions where, with sufficiently large turn rates, feasible flight is possible. As altitude increases, the gap of infeasible flight conditions becomes narrower and narrower as the lower density makes the adverse effects of the aileron easier to compensate.

The variation of the shape of the trim database with altitude, though, makes the choice of reasonable $\tilde{D}$ difficult. The method presented in Section 5.1 requires determining the intersection of the trim databases across altitude, thereby eliminating altitude as a degree-of-freedom. Given the dependence of trim state feasiblility, D' would be largely unpopulated if the intersection was taken from sea-level to 30,000 ft. To limit the effects of the altitude variation on $\tilde{D}$, intersections will be required to 10,000 ft in the following examples, enabling no dependence on altitude for landing trajectory generation. The shape of the resulting databases, though, result in partially connected transition maps severely

limiting flight capabilites.

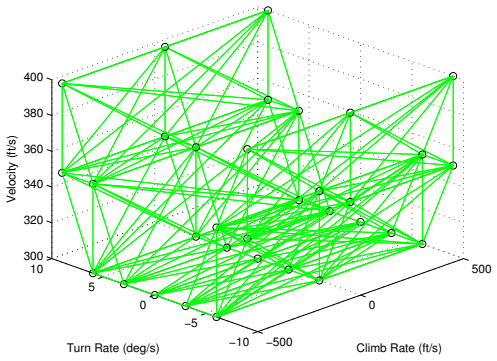
6.3.2 5° Aileron Jam Scenario

This example explores a 5° aileron jam, using the same initial location and desired landing site as previous examples. The $\tilde{D}$ used here must be carefully designed due to the U-shaped feasible trim database and will demonstrate the inherent trade-off in $\tilde{D}$. One could define $\tilde{D}$ as points with airspeeds 300 ft/sec and lower to allow straight and level flight and a fully-connected transition map. However, $\tilde{D}$ must include the airspeed where this failure occurs. Also, there may be urgency to land without further delay, given further uncertainty in failure cause.

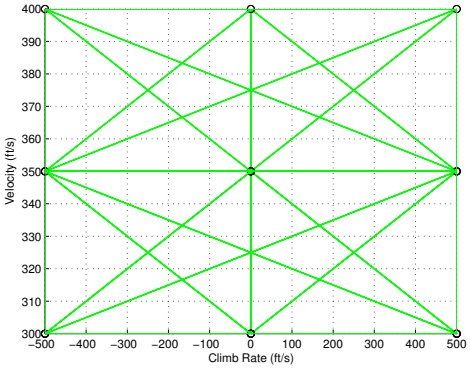
In the end, the chosen $\tilde{D}$ for this example failure scenario is shown in Figure 6.27 with the specific values used displayed in Table 6.13. Compiling $\tilde{M}$ was done using the values of $\Delta t = 20$ and $t_c = 10$ and mapping the displaced ground track over every thousand feet of altitude from sea-level to $10,000$ ft. Examining the resulting structure of $\tilde{M}$, the transition mapping was found to exclude transitions requiring more than a ±50 ft/s airspeed change due to throttle saturation. Those transitions that were found to feasible are represented in Figure 6.27.

Table 6.13: $\tilde{D}$ Values for 5° Aileron Jam

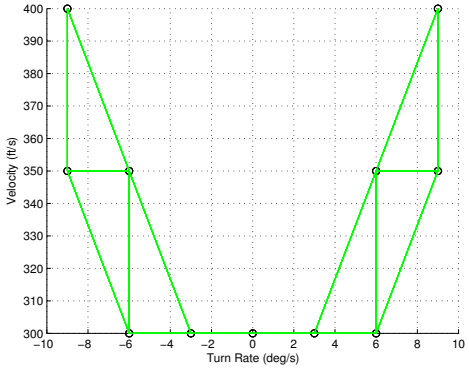
Airspeed (ft/s)	Climb Rates (ft/min)	Turn Rates (deg/s)	Points
300	0, ±500	0, ±3, ±6	15
350	0, ±500	±6, ±9	12
400	0, ±500	±9	3
Total Points			33



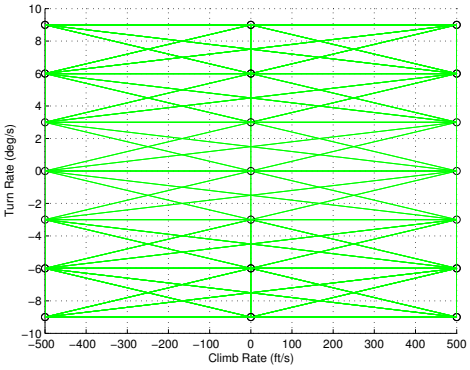
(a) 3D View



(b) x-z Projection



(c) y-z Projection



(d) x-y Projection

Figure 6.27: $\tilde{D}$ and $\tilde{M}$ for 5° Aileron Jam

Again, the simulated landing constraints of non-increasing airspeed segments and a final airspeed of $v_N = 300$ ft/sec were imposed to compute the five-segment optimal solution shown in Figure 6.28 and summarized in Table 6.14. The additional segment is required to compensate for the shape of $\tilde{D}$. In this example, multiple segments are needed to transition the aircraft to straight flight, a necessity to meet the terminal position constraints. The downside to increasing the number of segments used is in the expansion of the search space size. In this example, the extra segment increased the search space size by a factor of 33.

In the process of finding the optimal solution, the planner updated the optimal list five times as shown in Figure 6.29. Each progressively optimal solution shortened the two straight segments (segments 3 and 5), which resulted in the movement of the position of the fourth segment closer to the initial spiral of segments 1 and 2. To compensate for the increasing angle between the two straight segments, the planner lengthened the duration of the turns in segments 2 and 4. The planner could not make this trade-off, however, by directly iterating on the duration times, but by indirectly iterating over different sequences of climb rates. Because different climb rate sequences minimally effect the x and y position, the planner could substitute direct iteration on the duration times for iterations over the climb rate sequence. This iterative process would not be possible without a fifth segment.

Additionally, Table 6.16 shows increased time-to-solution data for these updated solutions. This increase is a direct effect of the shape of $\tilde{D}$ and the initial flight condition. Figure 6.30 shows the sparseness of solutions in the search space. The exhaustive search does not reach these promising regions until well into its iterations. If the layout of the solution densitycould be predicted *a priori*, the

Table 6.14: Optimal Plan for 5° Aileron Jam

i	v_i^* (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
0	400	0	−9	0.00
1	350	0	−6	16.487
2	300	−500	−3	34.504
3	300	500	0	1.070
4	300	−500	3	56.576
5	300	0	0	86.108
Total Plan Cost ΔT				194.744

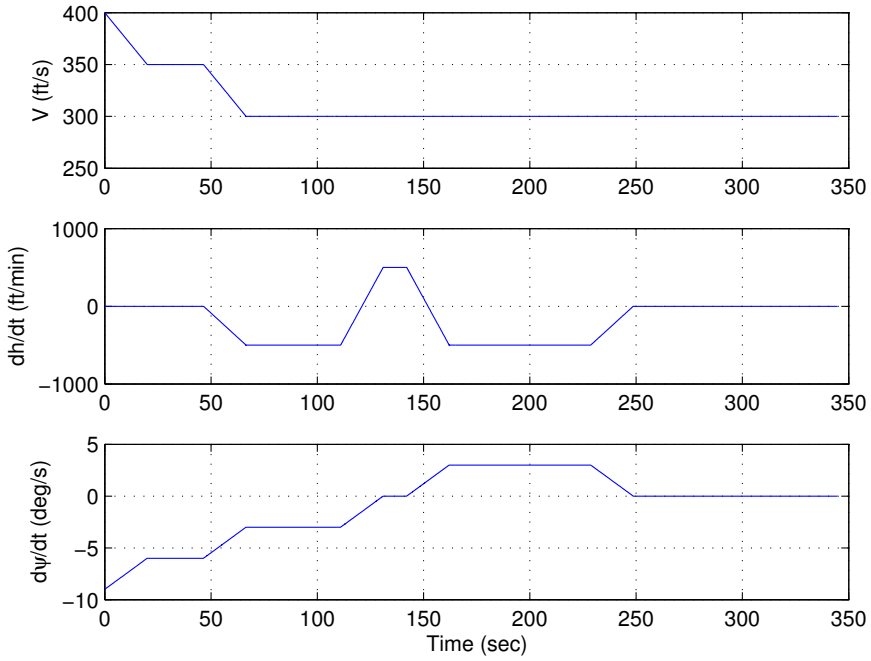
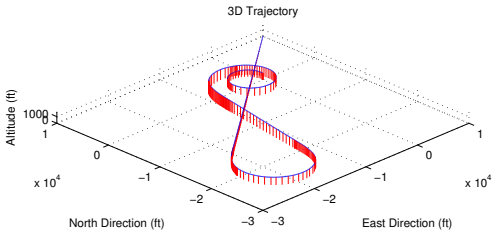
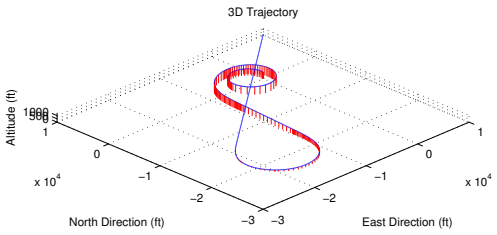


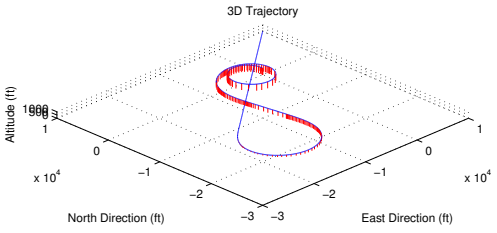
Figure 6.28: Optimal Plan for 5° Aileron Jam



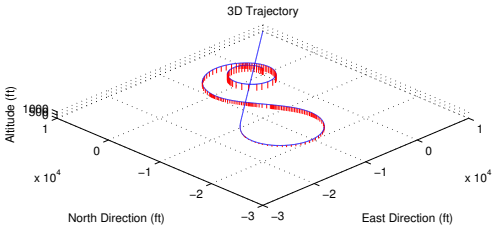
(a) First Solution



(b) Second Solution



(c) Third Solution



(d) Opitmal Solution

Figure 6.29: Solution Trajectories for 5° Aileron Jam

Table 6.15: Flight Path Errors for 5° Aileron Jam

Time (sec)	x_{error} (ft)	y_{error} (ft)	h_{error} (ft)	ψ_{error} (deg)	κ
30.000	−0.1024	−0.2139	−0.0000	0.0000	0.2372
76.487	0.0540	0.0283	−0.0005	0.0000	0.0609
140.990	0.3005	4.4951	0.0607	−0.0000	4.5055
172.060	0.7384	6.5219	0.0634	−0.0000	6.5639
258.637	0.2839	5.7062	0.0799	−0.0000	5.7138
344.744	0.6272	5.4267	0.0799	−0.0000	5.4634

discrete optimization could be trained on these candidate solutions first. Such a directed search algorithm may prove critical to real-time performance in future work.

A comparison between the optimal planned and simulated trajectories is shown in Table 6.15, with measurements again taken at the beginning of the trim segments. The results show that the actual trajectory diverges from the optimal plan between the second and third trim segments. This error is likely due to inaccuracies in the interpolation routine computing the transition ground track displacement. It is interesting to note that the error does not propagate to the altitude or heading due to the integral output tracking on climb rate and turn rate. Exploration into the various interpolation calculations is critical to futher decreases in planning errors in future work.

The evolution of the flight condition, displayed in Figure 6.31, shows similar transient behavior to the rudder failures, as well as near perfect tracking along the trim segments. Most of the additional errors observed for this aileron jam

Table 6.16: Solution Updates for 5° Aileron Jam

Update	Time-to-Solution (sec)	Solution Cost (sec)
1	330.40	243.59
2	701.60	216.13
3	702.14	195.26
4	702.72	194.74
Total	1266.07	194.74

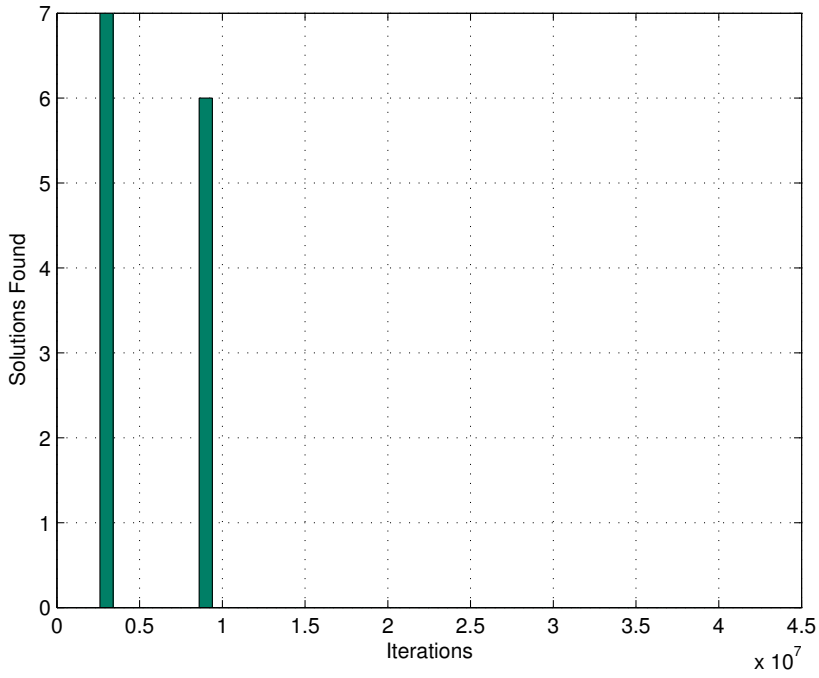


Figure 6.30: Solution Density for 5° Aileron Jam

scenario occur in the orientation (Figure 6.32) and required control inputs (Figure 6.33). Due to the relatively significant aerodynamic forces applied, the aileron is generally more effective at providing rolling moments than the rudder, which is clearly seen in the size of the large magnitude rudder deflections. However, even with this elevated actuation response, the control surfaces remain within their physical limits (see Section 2.3).

6.3.3 10° Aileron Jam Scenario

In this final example, the more severe 10° aileron jam is simulated. As with the 5° jam, the trim database is severely contracted. Through careful reduction, the $\tilde{D}$ used by the planner is shown in Figure 6.34 with Table 6.17 providing the specific flight condition values. However in this case, the databases $\tilde{T}$ and $\tilde{M}$ were calculated from sea level to 5,000 ft due to the lack of feasible flight conditions up to 10,000 ft. The values $\Delta t = 20$ s and $t_c = 10$ s were used to compile $\tilde{M}$. Again, the shape of the $\tilde{D}$ lead to numerous infeasible flight conditions in the reduced transition database; those that are feasible for this failure are shown in Figure 6.34.

The optimal solution for this example is given in Table 6.18 and shown graphically in Figure 6.35. To produce this solution, the initial aircraft location and desired landing site were the same as for all previous cases, but the number of segments was increased to $N = 6$. The additional segment allows the planner more freedom to move through $\tilde{D}$ when finding suitable sequences. This example, as well as the 5° $\tilde{D}$ structure, provides a simple strategy which the planner can use to determine the number of segments to use. Specifically, the planner can set N equal to four plus the minimal number of segments possible to connect the

Table 6.17: $\tilde{D}$ Values for 10° Aileron Jam

Airspeed (ft/s)	Climb Rates (ft/min)	Turn Rates (deg/s)	Points
216	0, ±500	0	3
232	0, ±500	±5, ±10	12
280	0, ±500	±10, ±12.5	12
328	0, ±500	±12.5	6
376	0, ±500	±15	6
Total Points			39

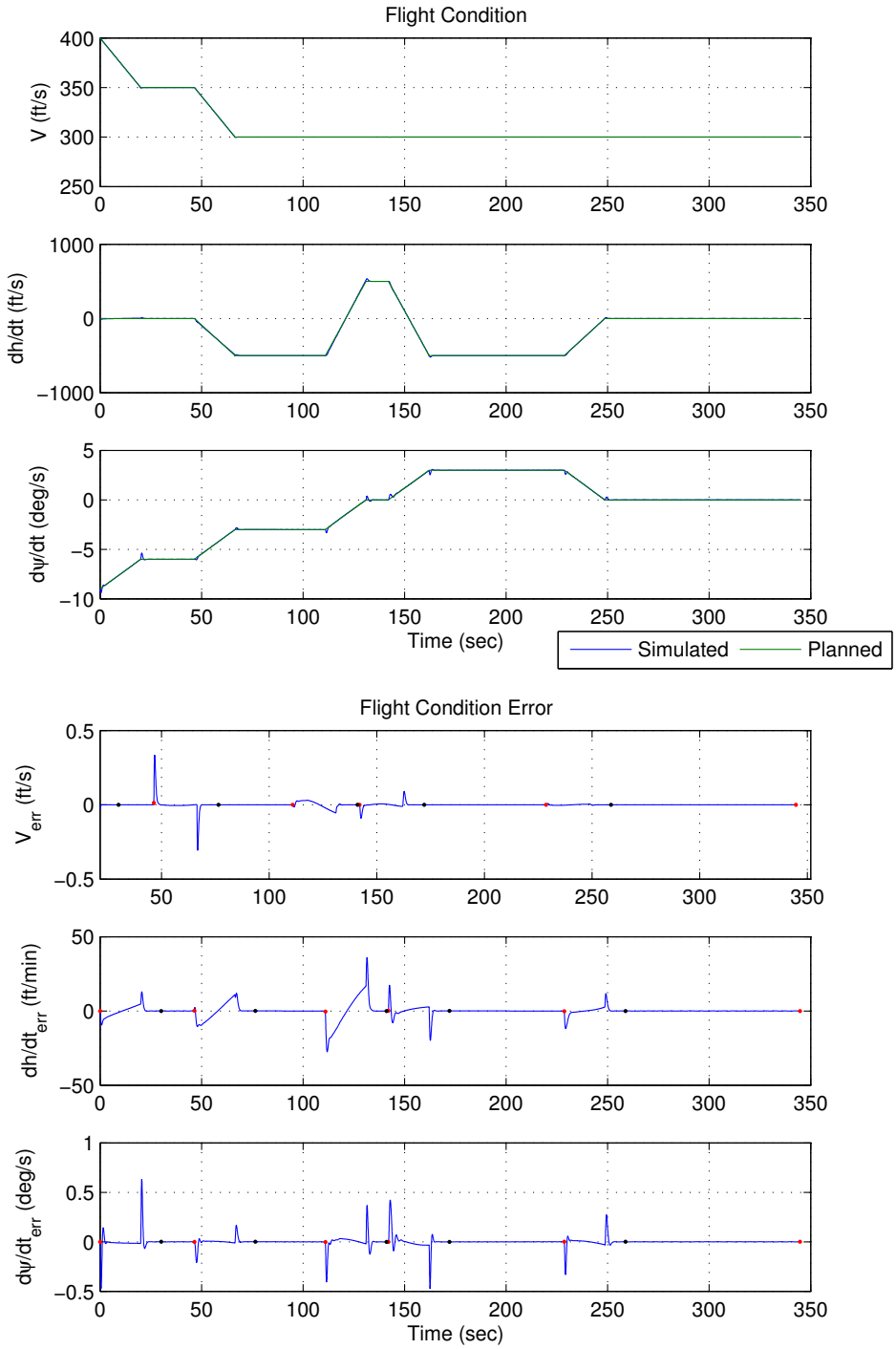


Figure 6.31: Flight Condition Information for 5° Aileron Jam

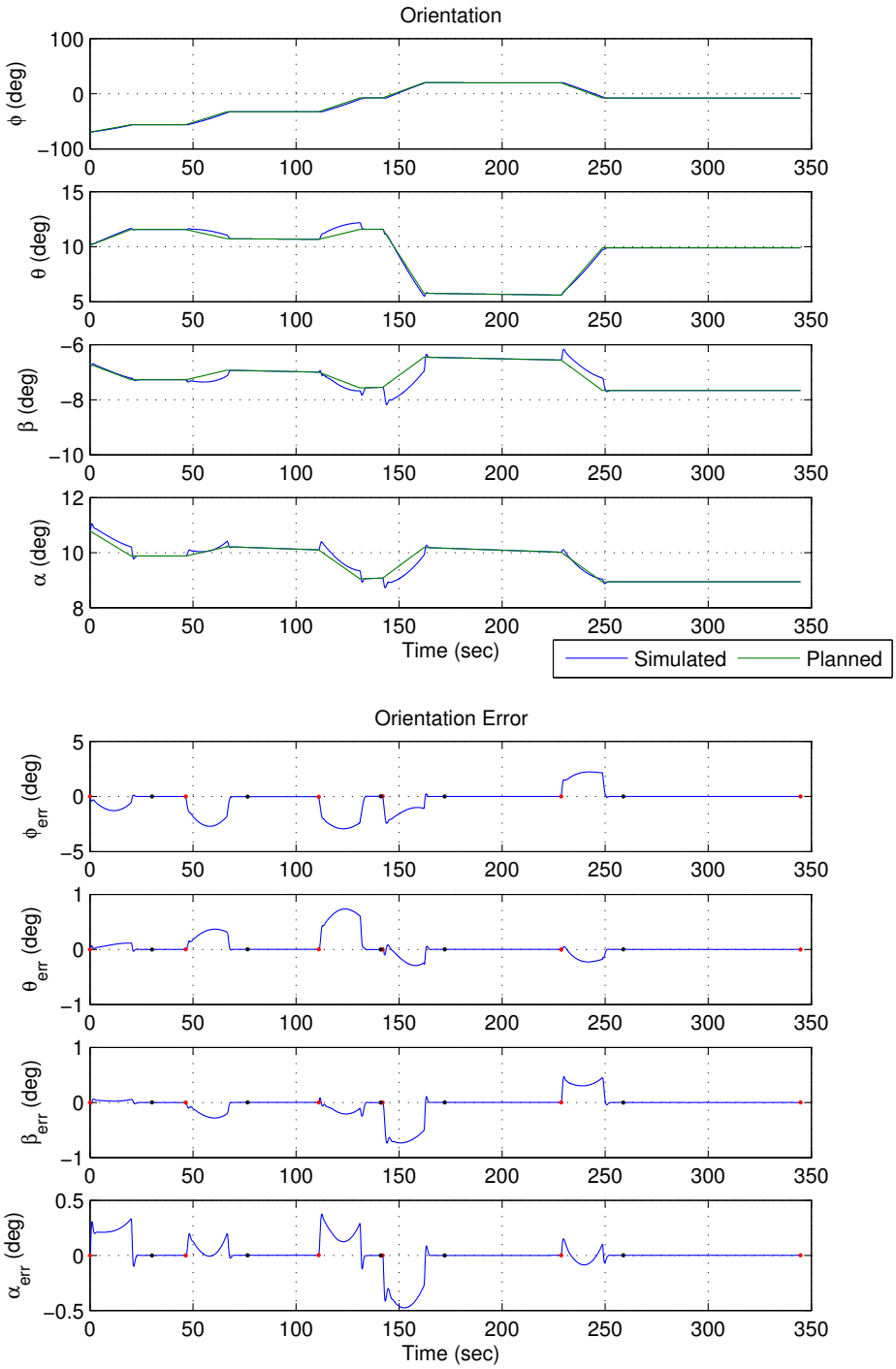


Figure 6.32: Orientation Information for 5° Aileron Jam

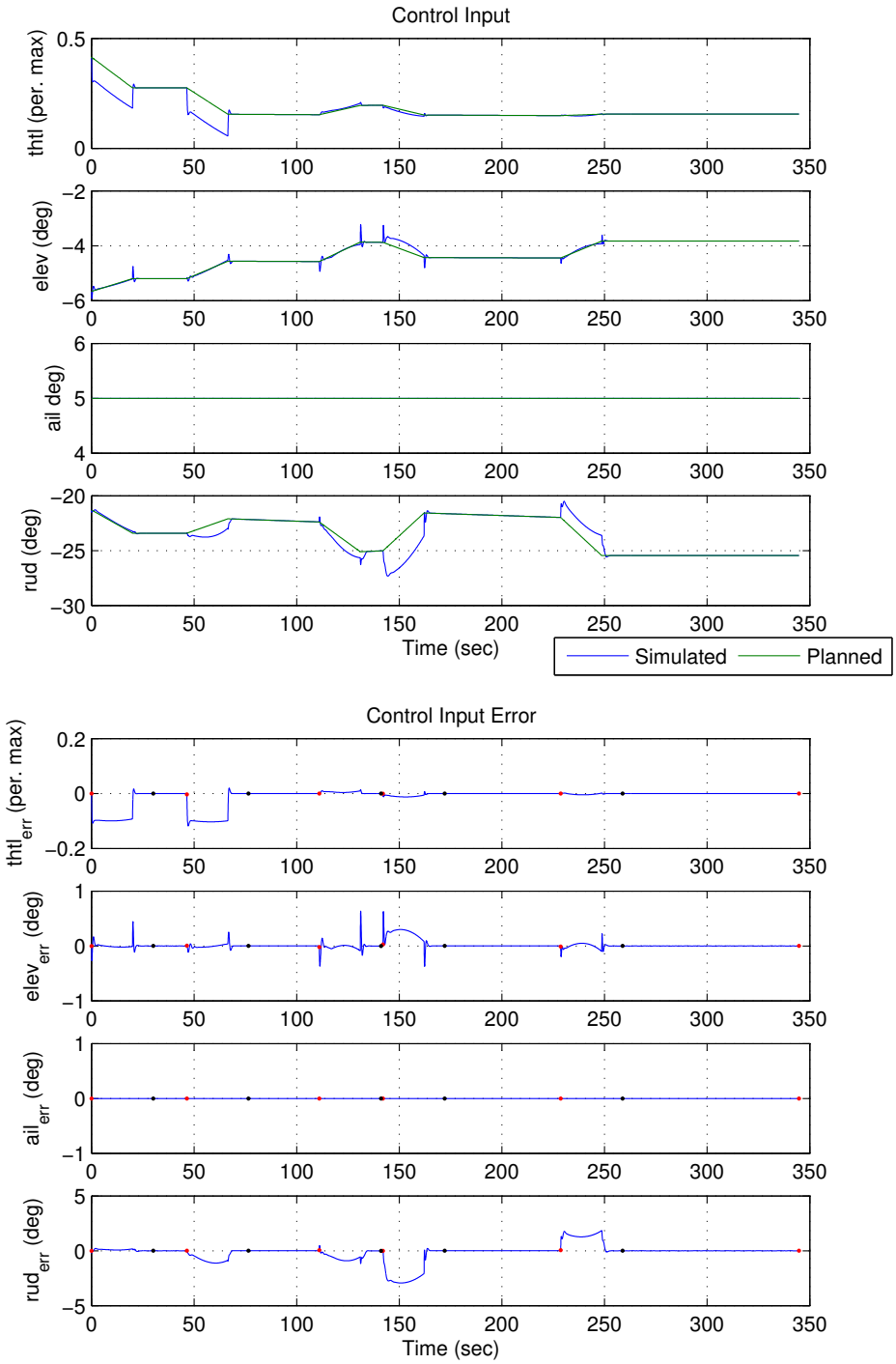
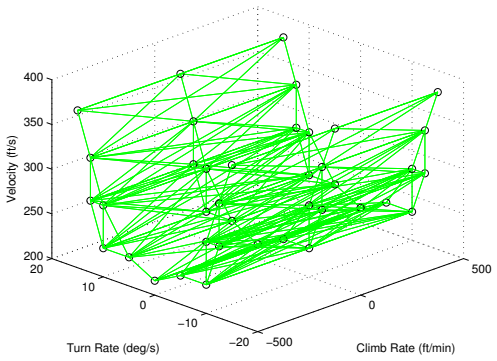
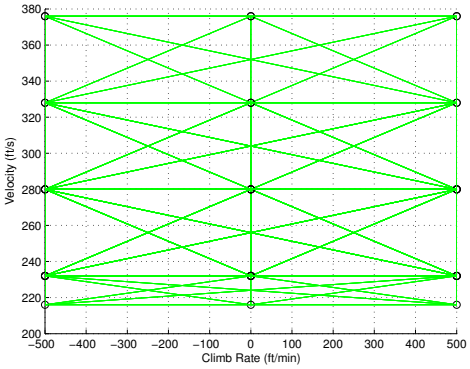


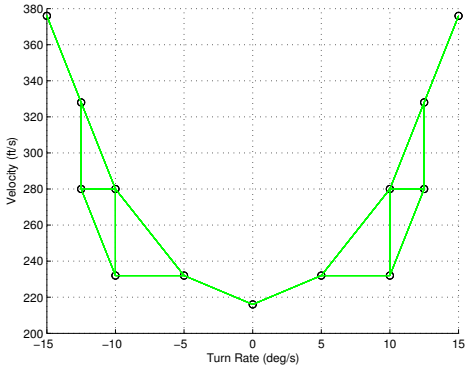
Figure 6.33: Controller Information for 5° Aileron Jam



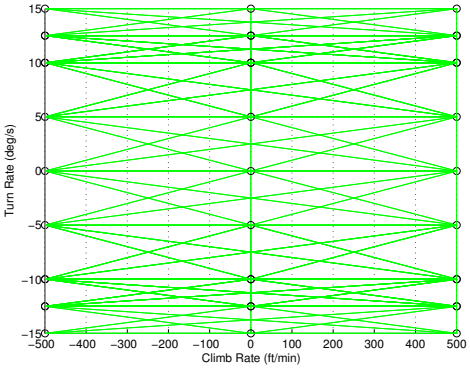
(a) 3D View



(b) x-z Projection



(c) y-z Projection



(d) x-y Projection

Figure 6.34: $\tilde{D}$ and $\tilde{M}$ for 10° Aileron Jam

Table 6.18: Optimal Plan for 10° Aileron Jam

i	v_i^* (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
0	376	−500	15	0.00
1	328	500	12.5	6.008
2	280	−500	10	3.427
3	232	−500	5	13.150
4	216	0	0	8.660
5	232	−500	5	39.434
6	216	0	0	5.157
Total Plan Cost ΔT				75.836

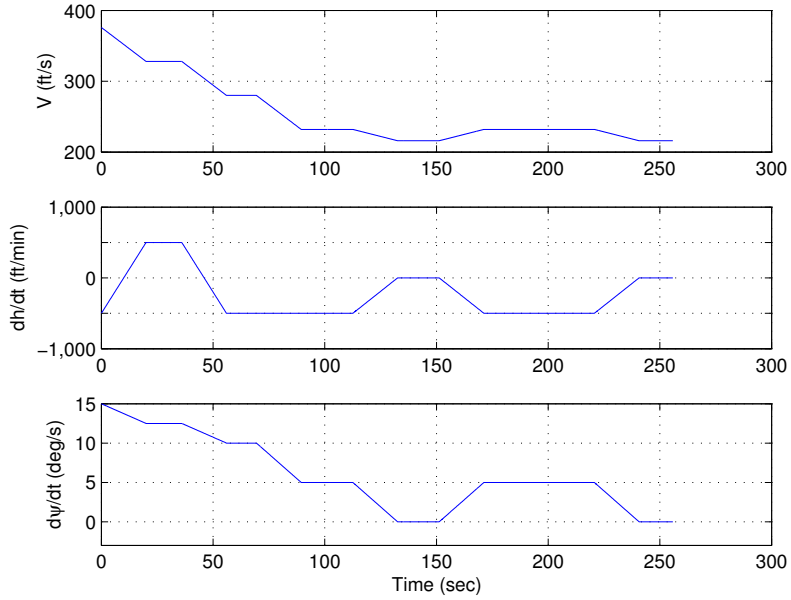


Figure 6.35: Optimal Plan for 10° Aileron Jam

initial flight condition and a straight flight condition.

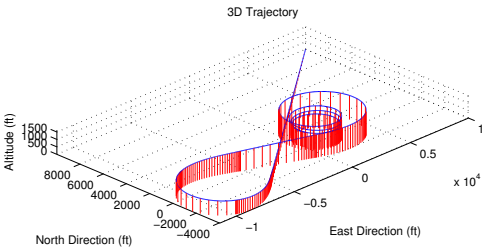
The optimal solution was again not the first found. The iterative update process is outlined in the trajectories shown in Figure 6.36. Notice the tight spiral trajectory at the beginning of each flight plan: because the planner has terminal airspeed and turn rate constraints, all acceptable solutions must move from the initial flight condition down the right section of the U-shaped $tildeD$, maintaining a positive turn rate over these segments. Additional segments allow the planner to choose over a variety of different final approaches to identify the optimal solution. The planner updates the optimal entry of the list eight different times as shown in Table 6.20. Notice the large reduction in cost between the first and second update but the minimal subsequent reduction. As with the 5° aileron jam, Figure 6.37 shows the sparseness of acceptable solutions found by the planner.

In spite of the severity of the 10° aileron jam, the error between the simulated optimal solution (Figure 6.36(a)) and that returned by the planner is extremely low, as summarized in Table 6.19. Most notably, the capability of the PID controller to track heading and altitude is clearly seen.

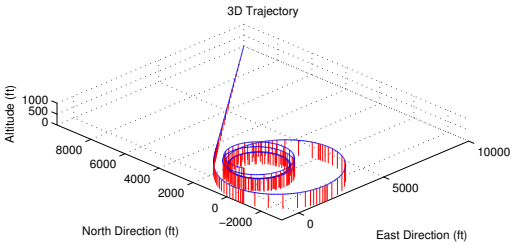
Again, Figure 6.38 shows the ability of the controller to track feasible flight condition sequences with minimal transient error. What makes the flight condition tracking performance much more impressive is to note the required aircraft configuration for these feasible trim states, shown in Figure 6.39. The severity of the 10° aileron jam requires extermely aggressive pitch and angle-of-attack values and pronounced side-slipping maneuvers. Additionally, feasible flight plans for this failure require complicated sequencing of the actuators, as seen in Figure 6.40, in order to maintain feasible flight conditions as well as produce necessary

Table 6.19: Flight Path Errors for 10° Aileron Jam

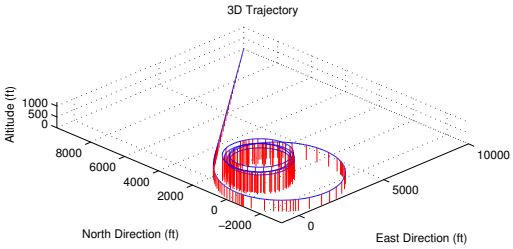
Time (sec)	x_{error} (ft)	y_{error} (ft)	h_{error} (ft)	ψ_{error} (deg)	κ
30.000	0.0241	−0.0080	−0.0009	0.0000	0.0254
66.008	0.2605	0.1031	−0.0044	0.0000	0.2802
99.435	−0.0527	−0.0331	−0.0043	0.0000	0.0624
142.586	1.6018	−0.8377	−0.0003	0.0000	1.3525
181.245	1.5508	−0.6764	0.0010	0.0000	1.6920
250.679	1.4156	−0.7610	0.0194	0.0000	1.6074
255.836	1.3902	−0.7366	0.0194	0.0000	1.5734



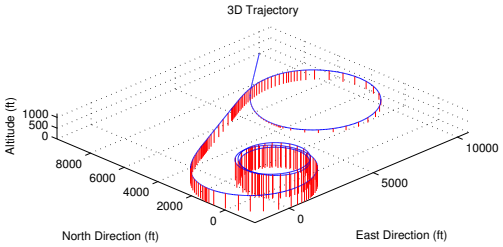
(a) First Solution



(b) Fourth Solution



(c) Sixth Solution



(d) Optimal Solution

Figure 6.36: Solution Trajectories for 10° Aileron Jam

Table 6.20: Solution Updates for 10° Aileron Jam

Update	Time-to-Solution (sec)	Solution Cost (sec)
1	192.65	170.45
2	248.19	79.40
3	300.03	77.78
4	304.88	77.34
5	483.60	76.66
6	485.10	76.56
7	738.45	76.40
8	1576.10	75.84
Total	17207.35	75.84

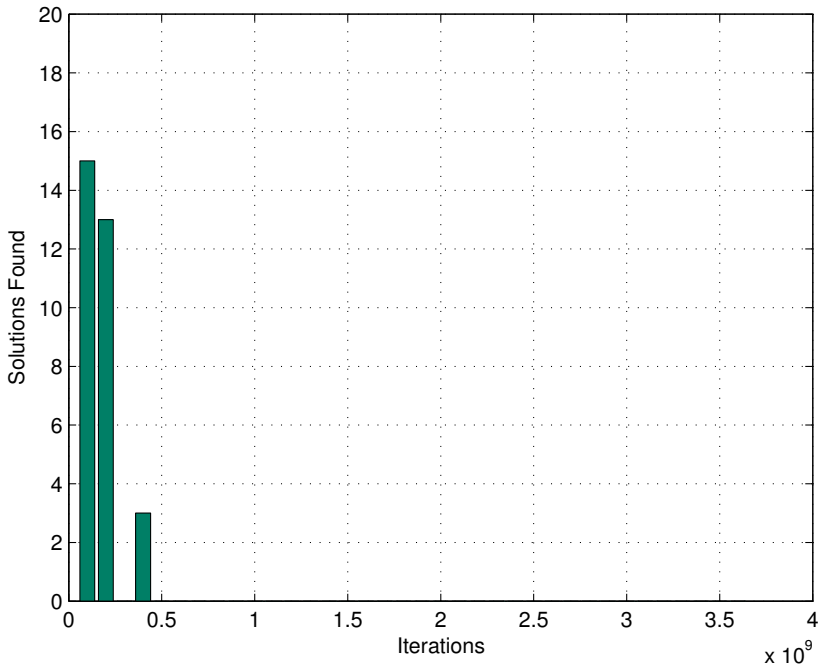


Figure 6.37: Solution Density for 10° Aileron Jam

transitions. Whereas less extreme rudder failures result in sufficiently large feasible flight envelopes to potentially allow pilots time to learn the relationship between non-intuitive bank and side slip configurations and dynamic responses, more severe failure scenarios, such as aileron jams, require quicker, more complex response that pilots could be expected to learn and execute.

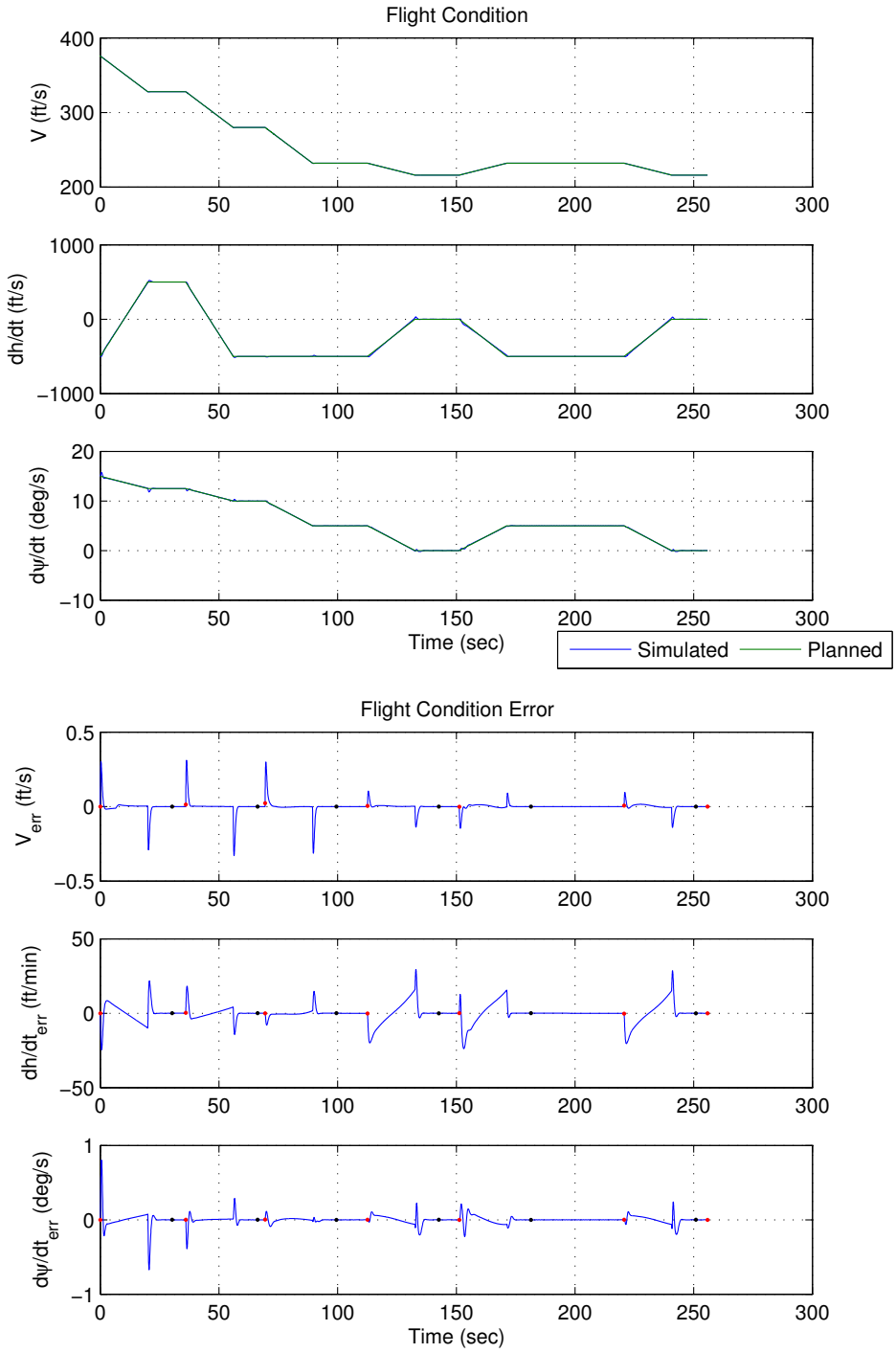


Figure 6.38: Flight Condition Information for 10° Aileron Jam

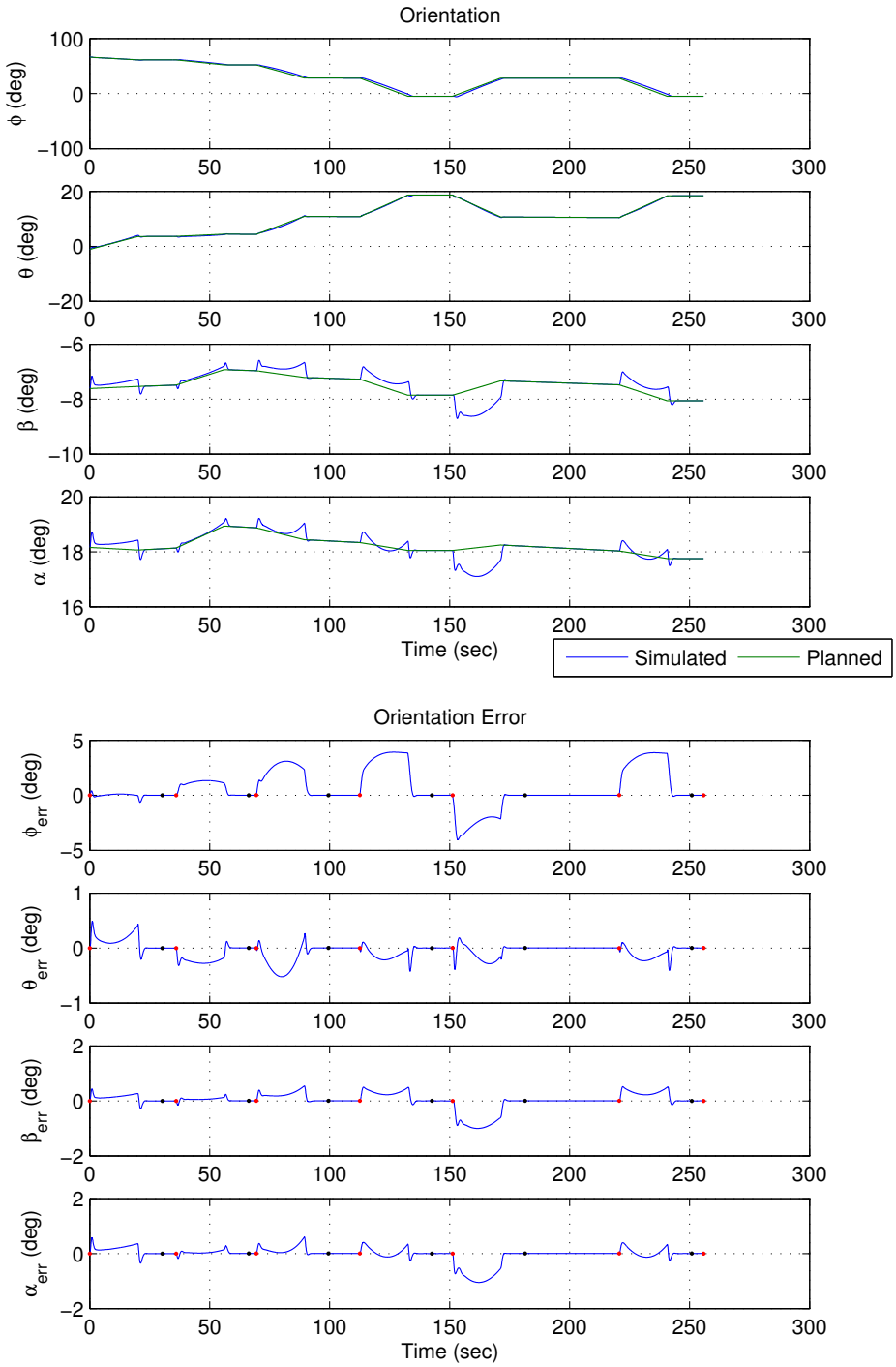


Figure 6.39: Orientation Information for 10° Aileron Jam

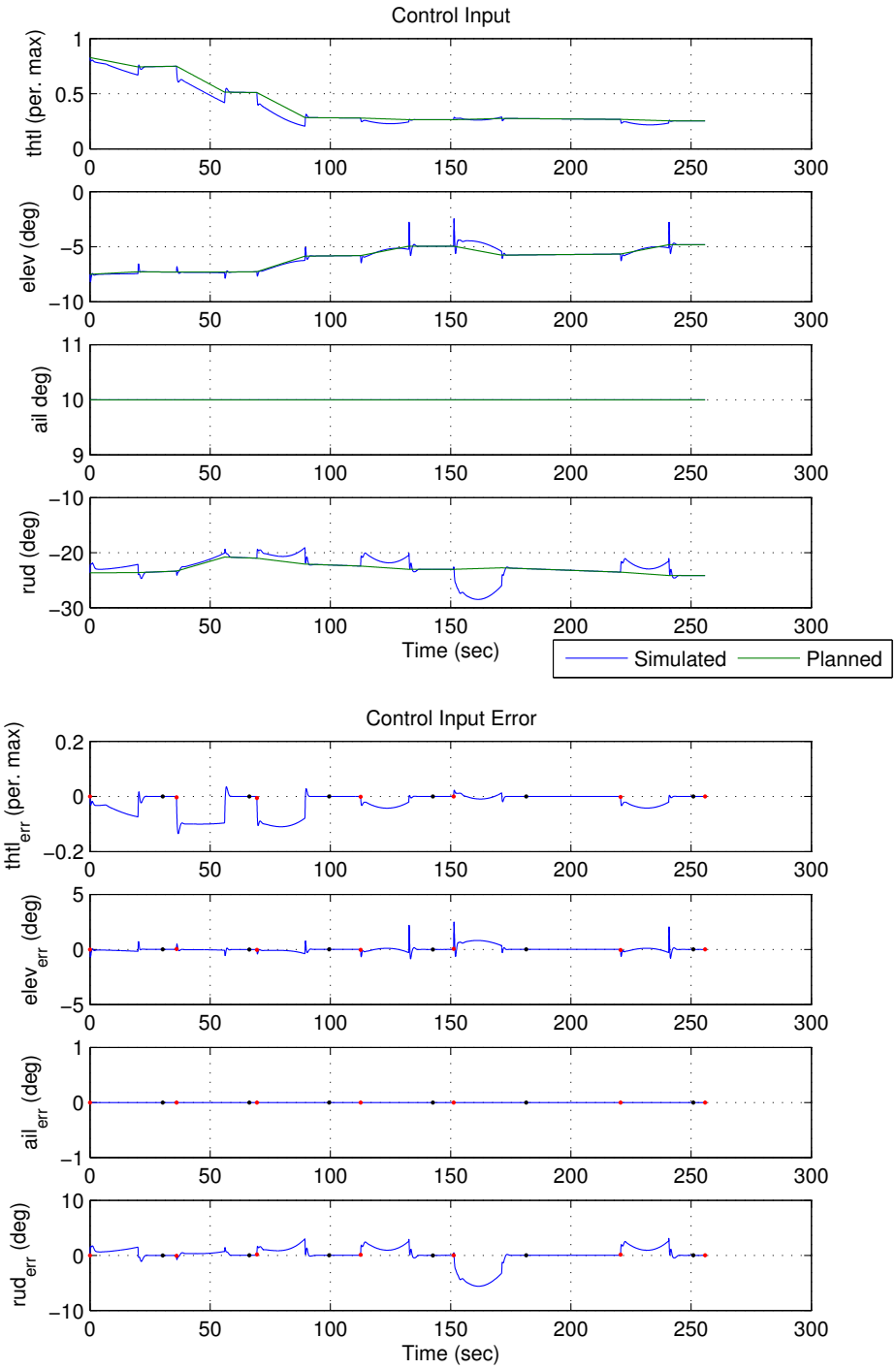


Figure 6.40: Controller Information for 10° Aileron Jam

Chapter 7

Conclusions and Future Work

Piloting an aircraft is a difficult task made more complex when emergency situations arise. When such emergencies result in significant reduction to the size and shape of the feasible flight envelope, effective response may challenge or even exceed most pilot's capabilities.

This thesis has described a method of autonomously generating feasible landing trajectories for aircraft in emergency situations that reduce flight performance. In particular, a method for computing post-failure flight envelopes from feasible trim states was presented and used to construct feasible trim state databases. Using a kinematic aircraft model, a trajectory planning algorithm was also defined which utilized the post-failure trim database to construct a feasible landing trajectory. A case study using an F-16 nonlinear model under varying degrees of rudder and aileron jams was presented to illustrate the utility of this approach.

The most important aspect of the planning strategy presented in this thesis is its generality. The use of trimmed flight conditions allows aircraft-specific performance information to be stored in database form. This analysis is performed off-line and can be refined before defining reduced trim and transition databases. When the planner is executed on-line, issues of flight envelope protection as well

as control limitations and saturations have already been addressed, allowing the planner to work with many different failure scenarios. Additionally, this generality allows many different post-failure control strategies to be implemented. So long as the controller used to produce the transition mappings is also used during the actual emergency, the results from the planner will accurately represent the actual motion of the aircraft.

The very power provided by trim database concept is also its one weakness. One has to exhaustively examine all failures before making claims concerning the "safety" of this system. This method may succeed in the quest for handling jammed actuators, but such success is less certain for more general failures (floating control surfaces, structural damage, fuel starvation, etc.) unless some higher-level "generalized" trim/transition analysis can be developed.

The method of trajectory planning proposed by this thesis will be a useful contribution to the aerospace community, especially in the area of aircraft safety. The inability of human pilots to fully and accurately characterize the impact of flight envelope changes due to a severe actuator failure has been the cause of many accidents. Indeed, the feasible trajectories found by the planning algorithm presented in this thesis for extreme cases of actuator failure result in very non-intuitive flight paths which nonetheless accurately navigate the vehicle to the specified landing site.

7.1 Future Work

7.1.1 Emergency FMS Integration

As mentioned in the introduction, an autonomous trajectory generator is only one piece of a larger emergency flight management system. One goal of future

research is to incorporate this planning algorithm into a full adaptive flight planner (AFP) as outlined in Figure 5.5. As mentioned, previous research has shown that landing site search (LSS) algorithms are capable of supplying the the autonomous trajectory planner a feasible landing site. The specific LSS algorithm uses a footprint calculation, based on the post-failure performance characteristics of the vehicle, to quickly determine the region reachable by the aircraft. The set of reachable runways—all airport runways within the footprint region—is then ranked according to a utility function containing terms such as the distance from the footprint boundary, runway length and width, wind speed and direction, and instrument approach quality. The current version of the LSS algorithm, however, has focused on engine-out failures for which the footprint calculation is straightforward and would need to be reformulated to account for more general failures, such as the control surface jams studied in this thesis.

Incorporation of the trim-based autonomous trajectory generation algorithm into an emergency flight planners (EFP) also requires the adoption of a standardized position reference, such as the global positioning system (GPS). Whereas this change is necessary to conform to commercial and general aviation standards, it requires the reevaluation of the equations used to derive the aircraft kinematic model. Implementation of the round-the-Earth equations of motion is possible, although future work would require the development of more sophisticated control systems.

7.1.2 Additional Failure Cases

Another area in which future work is being proposed is testing the planning scheme presented in this thesis on more diverse failure scenarios. In particular,

work is underway to test this strategy using a general commercial transport aircraft model missing a portion of its left wing. This structural change presents different challenges to an emergency flight planner. The loss of lifting area will most likely have a greater effect on flight envelope, contracting across both feasible turn and climb rates. Additionally, the failure model would retain its nominal control authority, which includes a series of redundant actuators such as spoilers, flaps, and the capability of differentially applying thrust from two different engines. With this additional control authority, the transition analysis becomes more interesting; increased control authority allows for more sophisticated control strategies resulting in better performance and potentially full configuration tracking.

In addition to the general transport model, future research should also focus on longitudinal actuator jams. Such failures have a greater potential to limit aircraft performance. The F-16 case studies from this work have shown that the control authority necessary to maintain trimmed flight conditions originates from throttle setting and elevator deflection. Therefore, one would expect a failure of either to drastically reshape the flight envelope. Indeed, the contraction of the flight envelope would be expected to simultaneously occur across both the feasible turn and climb rates. Furthermore, exploration of these failures may well lead to the discovery of "empty" trim databases. Such results are likely given the trim state definition presented in this thesis and would require the relaxation of the trim definition to allow "trim-like"—slightly accelerating—flight conditions. Previous work has utilized an alternate approach in defining trimmed flight segments based on energy conservation that allowed effective trajectory planning for engine-out scenarios [7].

It would also be interesting, and beneficial, to examine the effects of coupling different actuator failures together. Such failures would not only affect the set of feasible trim states, but also their properties. Coupling both lateral actuators together would most likely result in some set of feasible trim states. However, with no additional roll control authority, the aircraft would be unable to transition between these states.

7.1.3 Optimization Refinement

One final area of future research is in refining the mixed continuous/discrete optimization utilized by the trajectory planner. The goal of emergency trajectory planning is finding satisficing flight plans maneuvering the aircraft safely to a landing site. Accomplishing this task requires the implementation of a continuous optimization routine that effectively balances the trade-off between computational complexity and solution density. Currently, the Nelder-Mead simplex routine is a local optimization algorithm and, as such, there may be flight condition sequences capable of producing acceptable solutions which are not fully explored. Using different algorithms designed to explore larger regions of the solution space, such as mixed integer programming, simulated annealing, evolutionary algorithms, etc., would increase the probability that more of these solutions were found, though, the increase in computational complexity would be nontrivial. Alternatively, the probability of finding a solution sufficiently close to the global minimum using less computationally complex routines can be increased with an additional pre-processing step to compute educated initial guesses. (Currently, the initial durations for all trim segments are set to zero.) Therefore, future work should be focused on finding the best algorithm/initial guess combination that

provides the most effective method of balancing the computational complexity with solution density.

The discrete (exhaustive) search may also be improved in future work by directing the search to toward promising regions of the search space. With knowledge of regions more likely to contain acceptable solutions, the probability that the discrete optimization would find a feasible solution, or more optimal solutions, more quickly will certainly increase. Once these more promising regions have been explored, the search can be continued as time permits to further reduce solution cost.

BIBLIOGRAPHY

[1] Rosay, C. J., "Baghdad A300 Incident Discussion," Internal Document, Nov. 2004, Presentation by Airbus Chief Test Pilot.

[2] Liden, S., "The Evolution of Flight Management Systems," *Proc. of the IEEE Digital Avionics Systems Conference*, 1994, pp. 157–169.

[3] Fishbein, S. B., *Flight Management Systems: The Evolution of Avionics and Navigational Technology*, The Smithsonian Institution, Praeger, Connecticut, 1995.

[4] Chen, T. L. and Pritchett, A. R., "Development and Evaluation of a Cockpit Decision Aid for Emergency Trajectory Generation," *AIAA Journal of Aircraft*, Vol. 38, No. 5, Sept.-Oct. 2001, pp. 935–943.

[5] Alonso-Portillo, I. and Atkins, E. M., "Adaptive Trajectory Planning for Flight Management Systems," *Proc. of* 40^{th} *AIAA Aerospace Sciences Conference*, Reno, NV, Jan. 2002, (AIAA2002-1073).

[6] Boskovic, J. D. and Mehra, R. K., "An Integrated Fault Management System for Unmanned Aerial Vehicles," *Proc. of the* 2^{nd} *AIAA Unmanned Unlimited Conference*, San Diego, CA, Sept. 2003, (AIAA2003-6642).

[7] Atkins, E., Alonso-Portillo, I., and Strube, M., "Emergency Flight Planning Appplied to Total Loss of Thrust," Accepted for Publication in AIAA Journal of Aircraft.

[8] Strube, M., "Real Time Analysis of Automatic Trajectory Generator," Unpublished ENAE788I Class Assignment at the University of Maryland at College Park.

[9] Stewart, J. and Shuck, T., "Flight-Testing of the Self-Repairing Flight Control System Using the F-15 Highly Integrated Digital Electronic Control Flight Research Facility," Tech. rep., NASA Tech. Memorandum 101725, Aug. 1990.

[10] Wagner, E. A., "Onboard Automatic Aid and Advisory for Pilots of Conflict Resolution Maneuvers," *Journal of Guidance, Navigation, and Control*, Vol. 14, No. 4, Jul.-Aug. 1991, pp. 822–833.

[11] Rysdyk, R. and Calise, A., "Nonlinear Adaptive Flight Control Using Neural Networks," *IEEE Control Systems Magazine*, Vol. 18, No. 6, Dec. 1998.

[12] Bragg, M. G., Basar, T., Perkins, W. R., Selig, M. S., Voulgaris, P. G., Melody, J. W., and Sarter, N. B., "Smart Icing Systems for Aircraft Icing Safety," *Proc.* 40^{th} *AIAA Aerospace Sciences Meeting & Exhibit*, Reno, NV, Jan. 2002, (AIAA2002-0813).

[13] KrishnaKumar, K., Limes, G., Gundy-Burlet, K., and Bryant, D., "An Adaptive Critic Approach to Reference Model Adaptation," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Austin, TX, Aug. 2003, (AIAA-2003-5790).

[14] Kaneshige, J. and Gundy-Burlet, K., "Integrated Neural Flight and Propulsion Control System," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, Aug. 2001, (AIAA2001-4386).

[15] Gundy-Burlet, K., "Augmentation of an Intelligent Flight Control System for a Simulated C-17 Aircraft," *Journal of Aerospace Computing, Information, and Communication*, Vol. 1, No. 12, Dec. 2004, pp. 526–542.

[16] Boskovic, J. D., Prasanth, R., and Mehra, R. K., "A Multi-Layer Autonomous Intelligent Control Architecture for Unmanned Aerial Vehicles," *Journal of Aerospance Computing, Information, and Communication*, Vol. 1, No. 12, Dec. 2004, pp. 605–628.

[17] Schouwenaars, T., Mettler, B., Feron, E., and How, J., "Robust Motion Planning Using a Maneuver Automaton with Built-in Uncertainties," *Proc. of IEEE American Control Conference (ACC)*, Jun. 2003, pp. 2211– 2216.

[18] Mettler, B., Valenti, M., Schouwenaars, T., Kuwata, Y., and How, J., "Autonomous UAV Guidance Build-Up: Flight-Test Demonstration and Evaluation Plan," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Austin, TX, Aug. 2003, (AIAA2003-574).

[19] Valenti, M., Mettler, B., Schouwenaars, T., Feron, E., and Paduano, J., "Trajectory Reconfiguration for an Unmanned Aircraft," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, Aug. 2002, (AIAA2002-4674).

[20] Tomlin, C., Mitchell, I., and Ghosh, R., "Safety Verification of Conflict Resolution Maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 2, No. 2, Jun. 2001.

[21] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, 1998.

[22] Seywald, H., Cliff, E. M., and Well, K. H., "Range Optimal Trajectories for an Aircraft Flying in the Vertical Plane," *Journal of Guidance, Control and Dynamics*, Vol. 17, No. 2, Mar.-Apr. 1994, pp. 389–398.

[23] Seywald, H., "Long Flight-Time Range-Optimal Aircraft Trajectories," *Journal of Guidance, Control and Dynamics*, Vol. 19, No. 1, Jan.-Feb. 1994, pp. 242–244.

[24] Schultz, R. L., "Three-Dimensional Trajectory Optimization for Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 20, 1997.

[25] Slattery, R. and Zhao, J., "Trajectory Synthesis for Air Traffic Automation," *Journal of Guidance, Control, and Dynamics*, Vol. 20, 1997.

[26] Wu, S. and Guo, S., "Optimum Flight Trajectory Guidance Based on Total Energy," *Journal of Guidance, Control and Dynamics*, Vol. 17, No. 2, Mar.-Apr. 1994, pp. 291–296.

[27] Vormer, F., Mulder, M., Paassen, M., and Mulder, J., "Design and Preliminary Evaluation of a Segment-based Routing Methodology," *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, Aug. 2002, (AIAA2002-4861).

[28] Frazzoli, E., Dahleh, M. A., and Feron, E., "Robust Hybrid Control for Autonomous Vehicle Motion Planning," Tech. Rep. LIDS-P-2468, Massachusetts Institute of Technology, 1999, Revised version submitted to IEEE Trans. on Automatic Control.

[29] Frazzoli, E., Dahleh, M. A., and Feron, E., "Robust Hybrid Control for Autonomous Vehicle Motion Planning," *Proc.* 39^{th} *IEEE Conference on Decision and Control*, Sydney, Australia, Dec. 2000, pp. 821 – 826.

[30] Frazzoli, E., Dahleh, M. A., and Feron, E., "Maneuver-Based Motion Planning for Nonlinear Systems with Symmetries," Submitted as a regular paper to IEEE Transactions on Robotics and Automation.

[31] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, Wiley-Interscience, New York, 1992.

[32] Khalil, H., *Nonlinear Systems*, Prentice Hall, 3^{rd} ed., 2002.

[33] Chen, C.-T., *Linear System Theory and Design*, Oxford, Oxford, 3rd ed., 1999.

[34] Ogata, K., *Modern Control Engineering*, Prentice Hall, Upper Saddle River, NJ, 4^{th} ed., 2002.

[35] Craig, J., *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, 2^{nd} ed., 1989.

[36] Bryson, A. E. and Ho, Y. C., *Applied Optimal Control - Optimization, Estimation, and Control*, Hemisphere Publishing Corp., NW, Washington, D.C., 1975.

[37] Kirk, D. E., *Optimal Control Theory - An Introduction*, Prentice-Hall, Englewood Cliffs, NJ, 1970.

[38] Lewis, F. L. and Syrmos, V. L., *Optimal Control*, John Wiley & Sons, Inc., New York, 2^{nd} ed., 1995.

[39] Galassi et al, "GNU Scientific Library Reference Manual (2nd. Ed.)," ISBN 095416173.

[40] Strube, M., Sanner, R., and Atkins, E., "Dynamic Flight Guidance Recalibration After Actuator Failure," *Proc. of AIAA* 1^{st} *Intelligent Systems Technical Conference*, Chicago, Il, Sept. 2004, (AIAA-2004-6255).

[41] Boddy, M. and Dean, T. L., "Solving Time-Dependent Planning Problems," *Proc. of the* 11^{th} *International Joint Conference on Artifical Intelligence*, Detroit, Michigan, 1989, pp. 979–984.

[42] Russell, S. J. and Zilberstein, S., "Computing Real-Time Systems," *Proc. of* 12^{th} *International Joint Conference on Artifical Intelligence*, Sydney, Austrailia, Aug. 1991, pp. 212–217.

[43] Zilberstein, S., "Using Any-Time Algorithms in Intelligent Systems," *AI Magazine*, 1996.

[44] Skogestad, S. and Postlethwaite, I., *Multivariable Feedback Control: Analysis and Design*, John Wiley & Sons, West Sussex, England, 1st ed., 1996.